z/OS Communications Server
3.1

IP Messages: Volume 1 (EZA)

SC27-3654-60

Note:

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 1087.

This edition applies to 3.1 of z/OS® (5655-ZOS), and to subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2024-01-20

Contents

Figures....................v

About this document....................vii

Summary of changes for IP Messages: Volume 1 (EZA)....................xv

Chapter 1. IP message standards introduction.................... 1

Chapter 2. EZA0*xxxx* messages....................7

Chapter 3. EZA1*xxxx* messages....................53

Chapter 4. EZA2*xxxx* messages....................131

Chapter 5. EZA3*xxxx* messages....................359

Chapter 6. EZA4*xxxx* messages....................401

Chapter 7. EZA5*xxxx* messages....................533

Chapter 8. EZA6*xxxx* messages....................623

Chapter 9. EZA8*xxxx* messages....................817

Chapter 10. EZA9*xxxx* messages....................971

Chapter 11. EZAIN*xx* messages....................1055

Chapter 12. EZAOP*xx* messages....................1057

Appendix A. Additional information about remote procedure call (RPC) messages....................1063

Appendix B. Related protocol specifications....................1065

Appendix C. Accessibility....................1085

Notices....................1087

Bibliography....................1091

Figures

1. Sample IP message format... 1

2. Sample IP message identifier... 1

About this document

This document describes the Internet Protocol (IP) messages that occur in z/OS Communications Server. The information in this document supports both IPv6 and IPv4. Unless explicitly noted, information describes IPv4 networking protocol. IPv6 support is qualified within the text.

For information about how to set up, initialize, and customize your Transmission Control Protocol/Internet Protocol (TCP/IP) services system, see the z/OS Communications Server: IP Configuration Reference, the z/OS Communications Server: IP Configuration Guide and the z/OS Communications Server: IP Programmer's Guide and Reference. For information about how to use the applications on your TCP/IP system, see z/OS Communications Server: IP User's Guide and Commands.

This document refers to Communications Server data sets by their default SMP/E distribution library name. Your installation might, however, have different names for these data sets where allowed by SMP/E, your installation personnel, or administration staff. For instance, this document refers to samples in SEZAINST library as simply in SEZAINST. Your installation might choose a data set name of SYS1.SEZAINST, CS390.SEZAINST or other high level qualifiers for the data set name.

Who should read this document

This document assists TCP/IP operators, system programmers, and users to:

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

Familiarity with TCP/IP concepts and terms is assumed.

How this document is organized

The messages are listed in alphanumeric order by message ID. For each message ID, the books contains the text and a description of the message. This book contains the following chapters:

- Chapter 2, "EZA0xxxx messages," on page 7 contains messages in the EZA0*xxxx* range.
- Chapter 3, "EZA1xxxx messages," on page 53 contains messages in the EZA1*xxxx* range.
- Chapter 4, "EZA2xxxx messages," on page 131 contains messages in the EZA2*xxxx* range.
- Chapter 5, "EZA3xxxx messages," on page 359 contains messages in the EZA3*xxxx* range.
- Chapter 6, "EZA4xxxx messages," on page 401 contains messages in the EZA4*xxxx* range.
- Chapter 7, "EZA5xxxx messages," on page 533 contains messages in the EZA5*xxxx* range.
- Chapter 8, "EZA6xxxx messages," on page 623 contains messages in the EZA6*xxxx* range.
- Chapter 9, "EZA8xxxx messages," on page 817 contains messages in the EZA8*xxxx* range.
- Chapter 10, "EZA9xxxx messages," on page 971 contains messages in the EZA9*xxxx* range.
- Chapter 11, "EZAINxx messages," on page 1055 contains messages in the EZAIN*xx* range.
- Chapter 12, "EZAOPxx messages," on page 1057 contains messages in the EZAOP*xx* range.
- Appendix A, "Additional information about remote procedure call (RPC) messages," on page 1063 contains information about remote procedure call messages.
- Appendix B, "Related protocol specifications," on page 1065 lists the related protocol specifications for TCP/IP.
- Accessibility describes accessibility features to help users with physical disabilities.
- "Notices" on page 1087 contains notices and trademarks used in this document.

- "Bibliography" on page 1091 contains descriptions of the documents in the z/OS Communications Server library.

How to use this document

To use this document, you should be familiar with z/OS TCP/IP Services and the TCP/IP suite of protocols.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. See, How to send feedback to IBM for additional information.

Conventions and terminology that are used in this information

Commands in this information that can be used in both TSO and z/OS UNIX environments use the following conventions:

- When describing how to use the command in a TSO environment, the command is presented in uppercase (for example, NETSTAT).
- When describing how to use the command in a z/OS UNIX environment, the command is presented in bold lowercase (for example, **netstat**).
- When referring to the command in a general way in text, the command is presented with an initial capital letter (for example, Netstat).

All the exit routines described in this information are *installation-wide exit routines*. The installation-wide exit routines also called installation-wide exits, exit routines, and exits throughout this information.

The TPF logon manager, although included with VTAM®, is an application program; therefore, the logon manager is documented separately from VTAM.

Samples used in this information might not be updated for each release. Evaluate a sample carefully before applying it to your system.

z/OS no longer supports mounting HFS data sets (The POSIX style file system). Instead, a z/OS File System (ZFS) can be implemented. The term hierarchical file system, abbreviated as HFS, is defined as a data structure that has a hierarchical nature with directories and files. References to hierarchical file systems or HFS might still be in use in z/OS Communications Server publications.

Note: In this information, you might see the following Shared Memory Communications over Remote Direct Memory Access (SMC-R) terminology:

- RoCE Express®, which is a generic term representing IBM 10 GbE RoCE Express, IBM 10 GbE RoCE Express2, IBM 25 GbE RoCE Express2, IBM 10 GbE RoCE Express3, and IBM 25 GbE RoCE Express3 feature capabilities. When this term is used in this information, the processing being described applies to all of these features. If processing is applicable to only one feature, the full terminology, for instance, IBM 10 GbE RoCE Express will be used.
- RoCE Express2, which is a generic term representing an IBM RoCE Express2 feature that might operate in either 10 GbE or 25 GbE link speed. When this term is used in this information, the processing being described applies to either link speed. If processing is applicable to only one link speed, the full terminology, for instance, IBM 25 GbE RoCE Express2 will be used.
- RoCE Express3, which is a generic term representing an IBM RoCE Express3 feature that might operate in either 10 GbE or 25 GbE link speed. When this term is used in this information, the processing being described applies to either link speed. If processing is applicable to only one link speed, the full terminology, for instance, IBM 25 GbE RoCE Express3 will be used.
- RDMA network interface card (RNIC), which is used to refer to the IBM 10 GbE RoCE Express, IBM 10 GbE RoCE Express2, IBM 25 GbE RoCE Express2, IBM 10 GbE RoCE Express3, or IBM 25 GbE RoCE Express3 feature.

- Shared RoCE environment, which means that the "RoCE Express" feature can be used concurrently, or shared, by multiple operating system instances. The feature is considered to operate in a shared RoCE environment even if you use it with a single operating system instance.

Clarification of notes

Information traditionally qualified as Notes is further qualified as follows:

Attention
Indicate the possibility of damage

Guideline
Customary way to perform a procedure

Note
Supplemental detail

Rule
Something you must do; limitations on your actions

Restriction
Indicates certain conditions are not supported; limitations on a product or facility

Requirement
Dependencies, prerequisites

Result
Indicates the outcome

Tip
Offers shortcuts or alternative ways of performing an action; a hint

Prerequisite and related information

z/OS Communications Server function is described in the z/OS Communications Server library. Descriptions of those documents are listed in "Bibliography" on page 1091, in the back of this document.

Required information

Before using this product, you should be familiar with TCP/IP, VTAM, MVS™, and UNIX System Services.

Softcopy information

Softcopy publications are available in the following collection.

Titles	Description
IBM Z Redbooks	The IBM Z® subject areas range from e-business application development and enablement to hardware, networking, Linux®, solutions, security, parallel sysplex, and many others. For more information about the Redbooks® publications, see http://www.redbooks.ibm.com/ and http://www.ibm.com/systems/z/os/zos/zfavorites/.

Other documents

This information explains how z/OS references information in other documents.

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see z/OS Information Roadmap (SA23-2299). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, and also describes each z/OS publication.

To find the complete z/OS library, visit the z/OS library in IBM Documentation (https://www.ibm.com/docs/en/zos).

Relevant RFCs are listed in an appendix of the IP documents. Architectural specifications for the SNA protocol are listed in an appendix of the SNA documents.

The following table lists documents that might be helpful to readers.

Title	**Number**
DNS and BIND, Fifth Edition, O'Reilly Media, 2006	ISBN 13: 978-0596100575
Routing in the Internet, Second Edition, Christian Huitema (Prentice Hall 1999)	ISBN 13: 978-0130226471
sendmail, Fourth Edition, Bryan Costales, Claus Assmann, George Jansen, and Gregory Shapiro, O'Reilly Media, 2007	ISBN 13: 978-0596510299
SNA Formats	GA27-3136
TCP/IP Illustrated, Volume 1: The Protocols, W. Richard Stevens, Addison-Wesley Professional, 1994	ISBN 13: 978-0201633467
TCP/IP Illustrated, Volume 2: The Implementation, Gary R. Wright and W. Richard Stevens, Addison-Wesley Professional, 1995	ISBN 13: 978-0201633542
TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, W. Richard Stevens, Addison-Wesley Professional, 1996	ISBN 13: 978-0201634952
TCP/IP Tutorial and Technical Overview	GG24-3376
Understanding LDAP	SG24-4986
z/OS Cryptographic Services System SSL Programming	*SC14-7495*
z/OS IBM Tivoli Directory Server Administration and Use for z/OS	*SC23-6788*
z/OS JES2 Initialization and Tuning Guide	*SA32-0991*
z/OS Problem Management	*SC23-6844*
z/OS MVS Diagnosis: Reference	*GA32-0904*
z/OS MVS Diagnosis: Tools and Service Aids	*GA32-0905*
z/OS MVS Using the Subsystem Interface	*SA38-0679*
z/OS Program Directory	*GI11-9848*
z/OS UNIX System Services Command Reference	*SA23-2280*
z/OS UNIX System Services Planning	*GA32-0884*
z/OS UNIX System Services Programming: Assembler Callable Services Reference	*SA23-2281*
z/OS UNIX System Services User's Guide	*SA23-2279*
z/OS XL C/C++ Runtime Library Reference	*SC14-7314*
Open Systems Adapter-Express Customer's Guide and Reference	*SA22-7935*

Redbooks publications

The following Redbooks publications might help you as you implement z/OS Communications Server.

Title	Number
IBM z/OS Communications Server TCP/IP Implementation, Volume 1: Base Functions, Connectivity, and Routing	SG24-8096
IBM z/OS Communications Server TCP/IP Implementation, Volume 2: Standard Applications	SG24-8097
IBM z/OS Communications Server TCP/IP Implementation, Volume 3: High Availability, Scalability, and Performance	SG24-8098
IBM z/OS Communications Server TCP/IP Implementation, Volume 4: Security and Policy-Based Networking	SG24-8099
IBM Communication Controller Migration Guide	SG24-6298
IP Network Design Guide	SG24-2580
Managing OS/390 TCP/IP with SNMP	SG24-5866
Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender	SG24-5957
SecureWay Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements	SG24-5631
SNA and TCP/IP Integration	SG24-5291
TCP/IP in a Sysplex	SG24-5235
TCP/IP Tutorial and Technical Overview	GG24-3376
Threadsafe Considerations for CICS	SG24-6351

Where to find related information on the Internet

z/OS

This site provides information about z/OS Communications Server release availability, migration information, downloads, and links to information about z/OS technology

http://www.ibm.com/systems/z/os/zos/

z/OS Internet Library

Use this site to view and download z/OS Communications Server documentation

http://www.ibm.com/systems/z/os/zos/library/bkserv/

z/OS Communications Server product

The page contains z/OS Communications Server product introduction

https://www.ibm.com/products/zos-communications-server

IBM Communications Server product support

Use this site to submit and track problems and search the z/OS Communications Server knowledge base for Technotes, FAQs, white papers, and other z/OS Communications Server information

https://www.ibm.com/mysupport

IBM Communications Server performance information

This site contains links to the most recent Communications Server performance reports

http://www.ibm.com/support/docview.wss?uid=swg27005524

IBM Systems Center publications

Use this site to view and order Redbooks publications, Redpapers, and Technotes

http://www.redbooks.ibm.com/

z/OS Support Community

Search the z/OS Support Community Library for Techdocs (including Flashes, presentations, Technotes, FAQs, white papers, Customer Support Plans, and Skills Transfer information)

z/OS Support Community

Tivoli® NetView® for z/OS

Use this site to view and download product documentation about Tivoli NetView for z/OS

http://www.ibm.com/support/knowledgecenter/SSZJDU/welcome

RFCs

Search for and view Request for Comments documents in this section of the Internet Engineering Task Force website, with links to the RFC repository and the IETF Working Groups web page

http://www.ietf.org/rfc.html

Internet drafts

View Internet-Drafts, which are working documents of the Internet Engineering Task Force (IETF) and other groups, in this section of the Internet Engineering Task Force website

http://www.ietf.org/ID.html

Information about web addresses can also be found in information APAR II11334.

Note: Any pointers in this publication to websites are provided for convenience only and do not serve as an endorsement of these websites.

DNS websites

For more information about DNS, see the following USENET news groups and mailing addresses:

USENET news groups
comp.protocols.dns.bind

BIND mailing lists
https://lists.isc.org/mailman/listinfo

BIND Users

- Subscribe by sending mail to bind-users-request@isc.org.
- Submit questions or answers to this forum by sending mail to bind-users@isc.org.

BIND 9 Users (This list might not be maintained indefinitely.)

- Subscribe by sending mail to bind9-users-request@isc.org.
- Submit questions or answers to this forum by sending mail to bind9-users@isc.org.

The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS systems programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge

- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS

To access the z/OS Basic Skills Information Center, open your web browser to the following website, which is available to all users (no login required): https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zbasics/homepage.html?cp=zosbasics

Summary of changes for IP Messages: Volume 1 (EZA)

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Changes made in z/OS Communications Server 3.1

The following content is new, changed, or no longer included in z/OS 3.1.

Deleted information

EZA1535I (deleted in z/OS Version 2)
EZA5125I - EZA5574I (deleted in z/OS Version 2)
EZA5593I - EZA5708I (deleted in z/OS Version 2)

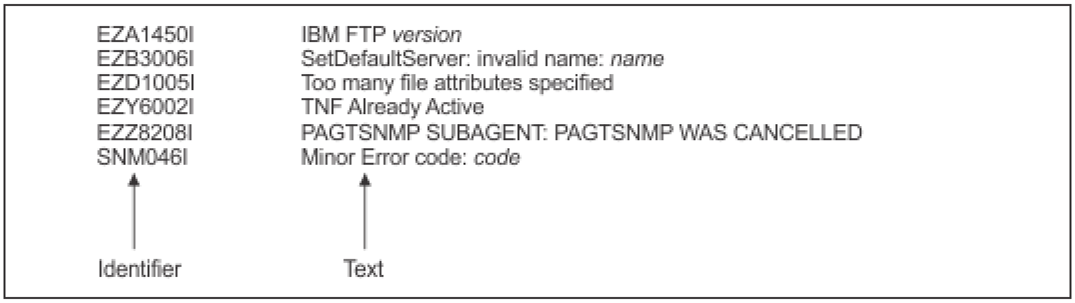
Chapter 1. IP message standards introduction

This topic contains the following information about IP message standards:

- "Message text formats" on page 1
- "Message description formats" on page 3
- "Message routing codes" on page 3
- "Message descriptor codes" on page 4
- "Message groups" on page 5

Message text formats

Most IP messages are preceded by an identifier, as illustrated in Figure 1 on page 1.



```
EZA1450I          IBM FTP version
EZB3006I          SetDefaultServer: invalid name: name
EZD1005I          Too many file attributes specified
EZY6002I          TNF Already Active
EZZ8208I          PAGTSNMP SUBAGENT: PAGTSNMP WAS CANCELLED
SNM046I           Minor Error code: code
```

Figure 1. Sample IP message format

Message identifiers

All message identifiers include the following sections:

- Prefix
- Message number
- Message type code

See Figure 2 on page 1 for a sample IP message identifier.

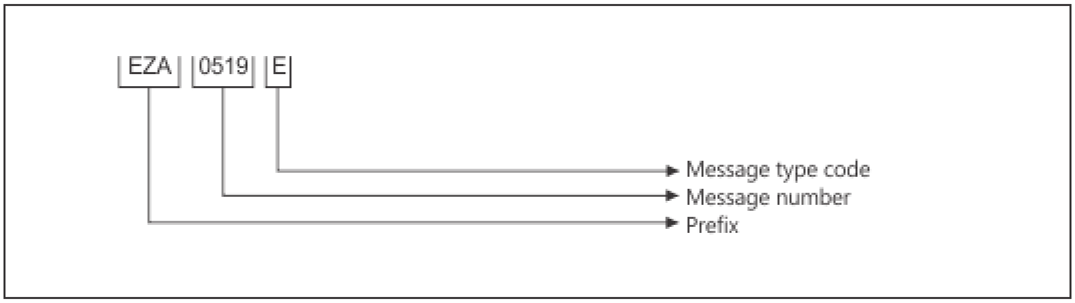


Figure 2. Sample IP message identifier

Prefix

Message identifiers include a prefix that identifies the source of the message. The following message prefixes are used by TCP/IP and its associated applications:

- EZA
- EZAIN

- EZAOP
- EZB
- EZBH
- EZD
- EZY
- EZYF
- EZYP
- EZYR
- EZYT
- EZYX
- EZZ
- SNM

Message number

Message identifiers include a unique 2- through 4-digit message number.

Message type code

The following type codes are used in IP messages:

A Action
The message indicates that an action is required.

E Eventual Action
You must eventually take some action to correct a problem. The system continues processing without waiting for a response.

I Information
The message is for your information. This type code can be used to notify you of an error. No response is necessary, but you might need to take some action.

S Severe Error
The message is for a system programmer.

W Wait
Processing stops until the operator takes a required action.

Syntax notation in message text

In this documentation, IP messages are described with the following syntax notation:

Non-highlighted characters
Represent the actual text of the message.

italic characters
Represent message variables. The variables are replaced by their values in the actual message.

Braces { }
Represent a group of text strings, only one of which is displayed in the actual message. The text strings are separated by or-signs (|) in the braces.

The braces and or-signs are not displayed in the actual message.

Brackets [ ]
Represent optional messages or optional parts of a message. Optional messages or optional parts of a message are displayed only under certain circumstances that are described in the "Explanation" section of the message. If an optional part has more than one possible value, or-signs separate the possibilities.

The brackets and or-signs are not displayed in the actual message.

Message description formats

A message consists of several sections. Not all sections are used for each message. For messages that are issued as a group, the "Explanation" section of the first message usually contains a complete description of the other messages in the group.

Explanation
Explains why the message was issued and describes all text and variables in the message.

System action
Explains the system state after the message was issued. This section also indicates whether the system is waiting for a reply.

Operator response
Describes actions that the operator can or must take at the console.

System programmer response
Suggests actions, programming changes, or system definition changes that isolate or correct errors or improve the efficiency of the system.

User response
Describes actions that the user can or must take at the terminal.

Problem determination
Additional instructions for determining the cause of the problem, searching problem databases, and if necessary, reporting the problem to the IBM support center. These instructions are for system programmers who can troubleshoot problems.

Source
Element, product, or component that issued the message.

Module
Module or modules that issued the message.

Automation
Indicates whether the message is a candidate for automation.

Example
Example of the message with variable fields replaced with actual values, perhaps in context with other messages.

Message routing codes

Routing codes determine where a message is displayed. More than one routing code might be assigned to the message. With multiple-console support, each console operator receives the messages related only to the commands entered at that console or to the functions assigned to that console, regardless of the routing codes assigned to those messages. If a message that is routed to a particular console cannot be issued at that console, that message is issued at the master console.

The following routing codes are used in IP messages:

Code
Meaning

1
Master Console Action: This message indicates a change in the system status and demands action by the master console operator.

2
Master Console Information: This message indicates a change in the system status. Such a message does not demand action, but alerts the master console operator to a condition that might require action. This routing code is used for any message that indicates job status, and also for processor and problem program messages to the master console operator.

3
: **Tape Pool**: This message specifies the status of a tape unit or reel, the disposition of a tape reel, or other tape-oriented information. For example, this can be a message which requests that tapes be mounted.

4
: **Direct Access Pool**: This message specifies the status of a direct access unit or pack, the disposition of a disk pack, or other direct-access-oriented information. For example, this can be a message which requests that disks be mounted.

5
: **Tape Library**: This message specifies the tape library information. For example, this can be a message which requests, by volume serial numbers, that tapes be obtained for system or programmer use.

6
: **Disk Library**: This message specifies the disk library information. For example, this can be a message which requests, by volume serial numbers, that disk packs be obtained for system or programmer use.

7
: **Unit Record Pool**: This message specifies the unit-record equipment information. For example, this can be a message which requests that printer trains be mounted.

8
: **Teleprocessing Control**: This message specifies the status or the disposition of data communication equipment. For example, this can be a message that indicates line errors.

9
: **System Security**: This message is associated with security checking. For example, this can be a message that requires a reply that is specifying a password.

10
: **System Error Maintenance**: This message indicates either a system error, or an input/output error that cannot be corrected. It also indicates a message that is associated with system maintenance.

11
: **Programmer Information**: This message is for the problem programmer. This routing code is used only when the program that issued the message cannot route the message to the programmer by using the system-output data set facility. The message is displayed in the system output message class of the job.

12
: **Emulators**: This message is issued by an emulator program.

13
: Reserved for customer use.

14
: Reserved for customer use.

15
: Reserved for customer use.

16
: Reserved for future expansion.

Message descriptor codes

Descriptor codes describe the kind of message being issued. These codes, with message routing codes, determine how a message is to be printed or displayed and how a message is to be deleted from a display device. Descriptor codes 1 – 7 are mutually exclusive; only one such code is assigned to a message. Descriptor codes 8 – 10 can be displayed with any other descriptor codes.

The following descriptor codes are used in IP messages:

Code
: **Meaning**

1
: **System Failure**: This message indicates that an error that cannot be corrected occurs. To continue, the operator must restart the system.

2
: **Immediate Action Required**: This message requires an immediate action by the operator. The action is required because the message issuer is in a wait state until the action is taken, or because system performance is degraded until the action is taken.

3
: **Eventual Action Required**: This message requires an eventual action by the operator. The task does not await completion of the action.

4
: **System Status**: This message indicates the status of a system task or of a hardware unit.

5
: **Immediate Command Response**: This message is issued as an immediate response to a system command. The completion of the response is not dependent on another system action or task.

6
: **Job Status**: This message contains status information regarding the job or job step.

7
: **Application Program/Processor**: This message is issued when a program is in problem mode.

8
: **Out-of-Line Message**: This message is one of a group of messages to be displayed out of line. If the device support cannot print a message out of line, the code is ignored, and the message is printed in line with other messages.

9
: **Request of the Operator**: This message is written in response to a request of the operator for information by the DEVSERV, MONITOR commands, and other operating system commands.

10
: This message is issued in response to a **TRACK** command.

11
: **Critical Eventual Action Required**: This message indicates that a critical event has occurred and must eventually be followed by an action. The message remains on the screen until the action is taken.

12
: **Important Information**: This message contains important information that must be displayed at the console, but does not require any action in response.

13–16
: Reserved.

Message groups

A message group contains two or more messages that are displayed together in response to a specific command or error condition. The following example is a message group.

```
EZZ8453I jobtype STORAGE

EZZ8454I jobname STORAGE     CURRENT MAXIMUM  LIMIT
EZD2018I location
EZZ8455I         storagetype current maximum  limit
EZZ8459I DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY
```

In most cases, the "Explanation" section of the first message in the group contains an example of the group and information about all messages in the group. The message descriptions of members of the group refer back to the first message for complete information.

Chapter 2. EZA0*xxxx* messages

EZA0519E **Unable to read dataset *dataset*, rc=*rc***

Explanation

TCPIP could not read the data set. The data set name is indicated along with a return code reason. There might be an earlier message that explains the error.

rc should be one of the following return codes. If *rc* is not one of the following return codes, then an unexpected return code was encountered.

1
Data set is empty

8
System error trying to obtain format 1 DSCB

12
Data set is migrated

16
Volume is not mounted

20
Volume is not direct access

24
Data set is VSAM

28
Invalid data set organization (DSORG) or record format (RECFM)

System action

MAKESITE halts. TCPIP continues.

Operator response

Check that correct access to the indicated data set is available before continuing. If *rc* is an unexpected return code, contact the System Programmer.

System programmer response

If *rc* is an unexpected return code, contact the IBM Support Center with the message.

Module

CMMAKSI

Procedure name

StartUp

EZA0521W **Warning: line *number* truncated to *number* characters**

Explanation

A line has been truncated because it is longer than the MaxLine constant. The MaxLine constant is the maximum number of characters that can be specified on a single entry line. Comments are ignored when determining the length of a line.

System action

TCPIP continues.

Operator response

If this truncation condition is unacceptable, correct the error before continuing. All entries to the HOSTS.LOCAL data set should be specified on a single line of up to a maximum of 512 characters per line. See z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

GetLine

EZA0522E **Line *line*: bad address syntax**

Explanation

An incorrect IP address was found in the input data set. The program procedure found a value different from the required IP address in dotted decimal notation.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

GetAddress

EZA0523E **Line *number*: only *number* fields**

Explanation

The indicated number of field terminators was less than the required minimum of three for the indicated line number in the input data set.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0524E **Line *number*: no type specified**

Explanation

The indicated line number in the input data set had a null type field.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0525E **Line *number*: unknown type *type***

Explanation

The indicated line number in the input data set was not null and did not contain HOST, GATEWAY, or NET in the type field.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0526E **Line *number*: no address specified**

Explanation

The indicated line number in the input data set did not contain a valid address.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0527W **Warning: line *number* has too many addresses; only the first *address* will be stored.**

Explanation

The data set contained more addresses than the number allowed by the indicated *MAXsiteADDRESSES* variable.

System action

TCPIP continues.

Operator response

Correct the error identified in the indicated line of the HOSTS.LOCAL data set and continue. See z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0528E **Line *number*: no name field**

Explanation

The indicated line number in the input data set did not contain a name field.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0529W **Warning: line *number* has too many names; only the first *name* will be stored.**

Explanation

The indicated line number in the input data set contained more names than the number allowed by the indicated *hbound(names)* variable.

System action

TCPIP continues.

Operator response

Correct the error identified in the indicated line and continue.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ParseLine

EZA0530E Line *number*: hash table overflow

Explanation

Data contained in the indicated line number of the input data set caused a hash table overflow because the *SiteTableSize* variable was exceeded. The size of the table entries is specified in the *userid.*HOSTS.LOCAL or the HOSTS.LOCAL data set. This message is followed by another message indicating that this is an irrecoverable error condition.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line of the HOSTS.LOCAL data set and continue. The maximum length for a host entry allowed in the HOST tables is 24 characters. However, the name server does not have maximum character length. If the error persists, contact the IBM Software Support Center.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

LookUp

EZA0531I H *value*, Tries *number*, Pos *address*, *address*, K *value*

EZA0532I Index@*address*, *address*, K *address*

Explanation

These messages provide additional information values after a table overflow has occurred. These messages are written to an output file and are displayed with EZA0530E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

See EZA0530E for more information.

Module

CMMAKSI

Procedure name

LookUp

EZA0533S ****** Fatal error: MakeSite aborted**

Explanation

This message follows EZA0530E which explains the specific cause of the error.

System action

MAKESITE halts. TCPIP continues.

Operator response

Correct the error identified in the indicated line and continue. The maximum length for a host allowed in the HOST tables is 24 characters. However, the name server does not have maximum character length. If the error persists, contact the IBM Software Support Center.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

LookUp

EZA0534W **Warning: *name1* agrees with *name2* in the first *CMSword_limit* characters**

Explanation

This message indicates a possible PseudoDuplicate name condition. This message is issued only if the WarnMe flag has been set to true with the WARN option. PseudoDuplicate indicates that the first displayed number of characters of the first host name are the same as the first number of characters of the second host name.

System action

TCPIP continues.

Operator response

If this condition is unacceptable, correct the error in the HOSTS.LOCAL data set before continuing.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

LookUp

EZA0535E **Address hash table overflow on host**

Explanation

The procedure MergeNames, which takes host names from the old HOST data set and adds them to the new HOST data set, was not successful because the *AddrTableSize* variable was exceeded for the host address hash table. This message is displayed with EZA0538W.

System action

TCPIP continues.

Operator response

Correct the error identified on the indicated line and continue. The maximum length for a host allowed in the HOST tables is 24 characters. If the error persists, contact the IBM Software Support Center.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

Insert

EZA0537W **Warning name lost: *name***

Explanation

The indicated host name was lost during the generation of the *user_id*.HOSTS.SITEINFO and the *user_id*.HOSTS.ADDRINFO data sets because the length of the host name was longer that the field specified. The maximum length for a host allowed in the HOST tables is 24 characters. This message is issued only if the WarnMe flag has been set to true with the WARN option.

System action

TCPIP continues.

Operator response

Reenter the host name in the HOSTNAME parameter of the HOSTS.LOCAL data set and restart the process. See z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

ProcessAdr

EZA0538W **Warning name lost: *name***

Explanation

The indicated name was lost during program processing. This message is issued only if the WarnMe flag has been set to true with the WARN option. This message is displayed with EZA0535.

System action

TCPIP continues.

Operator response

See message number EZA0535W for information.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MergeNames

EZA0539I **Merge A into B:**

Explanation

The operation performed a merge between the new address of A and existing addresses of B to form a current index.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0540I **A =**

Explanation

This message precedes the display of new addresses to be added to the current index.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0542I **B =**

Explanation

This message displays the old addresses that are currently present in the index. The address information from A will be merged with the addresses of B to produce a current data set index.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0543I **Computing tables ...**

Explanation

The site tables are now being executed for the MAKESITE program.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0544I **Merged current into Index *index* = *name*.**

Explanation

This message gives the current Index number and the name of the index.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0545I **Line *line*: the name '*names*' is a duplicate.**

Explanation

This message displays the addresses contained in the site table that are duplicate entries. The line number indicates the destination of the duplicate name.

System action

TCPIP continues.

Operator response

To correct the problem, edit the HOST.LOCAL data set. Discard any duplicate names and reissue MAKESITE.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0546I **Merged Index *index* = *name* into Current**

Explanation

The current index number and name are to be added to the current index.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0547I **Writing out *datasetname***

Explanation

The MAKESITE program is generating the data set name specified in the message text. For more information see z/OS Communications Server: IP Configuration Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0548I **Writing out *datasetname***

Explanation

The MAKESITE program is formatting information to the data set specified in the message text.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMAKSI

Procedure name

MainProg

EZA0549I **% S T A T I S T I C S**

Explanation

This message is a header for the specific statistics to follow. This message will be followed by a series of messages indicating the HOSTS.LOCAL, HOSTS.SITEINFO, and HOSTS.ADDRINFO data sets.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0550I **Dataset: *datasetname***

Explanation

This message header is followed by statistics describing the data set specified in the message text.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumoStatistics

EZA0551I **Total lines: *lines***

Explanation

This message indicates the total number of lines in the HOSTS.LOCAL data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0552W **Bad lines: (skipped) *duplicates***

Explanation

This message shows the number of lines that are not added to the HOSTS.LOCAL data set, because these lines are either duplicates or not readable.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0553I **Duplicate names: *duplicates***

Explanation

This message indicates the number of address names conflicting with address names that exist in the data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0554I **Conflicts in first *8* letters: *duplicates***

Explanation

If conflicts or duplicates occur in the first 8 letters, this message will be displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0555I ***total* networks, *total* gateways, *total* hosts**

Explanation

This message shows how many networks, gateways, and hosts are defined in your HOSTS.LOCAL data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0556I **Dataset: *name***

Explanation

This header message will be followed by other messages with specific details concerning the data set name specified in the message text.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0557I	**% Table Size: *size*.**
EZA0558I	**% Total Entries: *entries*.**
EZA0559I	**% Distinct Names: *names*.**
EZA0560I	**Collisions: *collisions*.**
EZA0561I	**Average probes/name: *1.0 + collisions/namesfound*.**

Explanation

These messages are displayed when the MAKESITE command is invoked. Each title states the HOSTS.SITEINFO data set required in your HOSTS.SITEINFO data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0562I	**Dataset: *FullDataSetName***
EZA0563I	**Table Size: *AddrTableSize***
EZA0564I	**Total entries: *AddrInserts***
EZA0565I	**Collisions: *AddrCollisions***
EZA0566I	**Names dropped: *AddrNamesLost***

Explanation

Message EZA0562I is followed by other messages with specific details concerning the data set name specified in the message text. Each message provides an indication of the setting for your HOSTS.ADDRINFO data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0567I	**MVS TCP/IP Makesite**

Explanation

This message identifies the operating system and the application.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0568W	**Size of SITE table exceeds the capacity of MAKESITE.**

EZA0569W	**Maximum size of SITE table:** ***number***

EZA0570W	**Computed SITE table size:** ***size*.**

Explanation

The size of the HOSTS.SITEINFO must be within the range of the MAKESITE program. Each entry should be specified on a single line of no more than 512 characters. The sizes of the tables currently in the HOSTS.SITEINFO data sets are shown. For more information on the HOSTS.LOCAL data set, see the z/OS Communications Server: IP Configuration Guide.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

StartUp

EZA0571W	**Size of Addr table exceeds the capacity of MAKESITE.**
EZA0572W	**Maximum size of ADDR table: *number***
EZA0573W	**Computed ADDR table size: *size***

Explanation

The size of the HOSTS.ADDINFO must be within the range of the MAKESITE program. Each entry should be specified on a single line of no more than 512 characters. The sizes of the tables currently in the HOSTS.ADDINFO data sets are shown.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

StartUp

EZA0574E	**MAKESITE Terminating Abnormally.**

Explanation

Data contained in the SITE table or the ADDR table exceeded the capacity of the MAKESITE program and caused an error.

System action

MAKESITE halts.

Operator response

Edit the HOSTS.LOCAL data set. Once you have access, edit the data set line number and name correctly and within range of the MAKESITE program. For more information on MAKESITE see z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

DumpStatistics

EZA0575I **MakeSite Parameters passed - *parameters*.**

Explanation

This message shows the parameters passed by the MAKESITE program. If the parameters are not correct, an error message will follow this message.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

StartUp

EZA0576E **MakeSite Parameter ERROR - *error*.**

Explanation

The parameter you have selected was not recognized by the MAKESITE program.

error is the unrecognized parameter.

System action

TCPIP continues.

Operator response

MAKESITE does not allow continuation lines; entries should be made on a single line of no more than 512 characters. Reissue the command using a valid parameter. For more information on MAKESITE parameters see z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

ParmExtract

EZA0578E **Output data set file allocation ERROR: *dataset***

Explanation

The allocation of the HOSTS.SITEINFO data set failed. The following causes are typical causes of this failure:

- An existing data set cannot be opened for modification because another user has it allocated.
- A new data set cannot be allocated because there is not sufficient space on the volume.
- The user does not have the authority to create the data set.

System action

MAKESITE halts.

Operator response

Notify the system programmer to correct the error.

System programmer response

Ensure that another user does not have the HOSTS.SITEINFO data set open for modification and that there is space available on the volume to create or expand the HOSTS.SITEINFO data set. Also ensure that the user has sufficient authority to access or create the HOSTS.SITEINFO data set.

Module

CMMASKI

Procedure name

StartUp

EZA0579E **Proceed Flag is OFF --HALT--.**

Explanation

This message will display if the data set specified in message EZA0578E is incorrect. This causes the procedure to discontinue until the error is corrected.

System action

MAKESITE halts.

Operator response

Make sure the syntax of the data set is correctly specified and MAKESITE is able to read the HOSTS.LOCAL data set.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

getstate

EZA0580I **Proceed Flag is ON -- Using default attributes.**

Explanation

This message indicates the procedure has continued and default attributes are being initialized. For more information about MAKESITE, see z/OS Communications Server: IP Configuration Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMASKI

Procedure name

StartUp

EZA0581I ***testname***

Explanation

Indicates the gateway, host, or net name for which a TESTSITE command was entered.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

PrintSite

EZA0591I **Name:**

Explanation

Prompts you to enter the gateway, host, or net name for which you need to enter the TESTSITE command.

System action

The TESTSITE program waits for your input. TCPIP continues.

Operator response

Enter the gateway, host, or net name to be verified.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

DoIt

EZA0592I **No host named *host***

Explanation

There are no HOST entries for the specified name in the HOSTS.ADDRINFO and *user_id.v3r1.*HOSTS.SITEINFO data sets.

System action

TCPIP continues and prompts the user for the next name.

Operator response

Correct the HOSTS.LOCAL data set and rebuild the HOSTS.ADDRINFO and *user_id.v3r1.*HOSTS.SITEINFO data sets with MAKESITE, as necessary.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

DoIt

EZA0593I **No gateway named *gateway***

Explanation

There are no GATEWAY entries for the specified name in the HOSTS.ADDRINFO and *user_id.v3r1.*HOSTS.SITEINFO data sets.

System action

TCPIP continues and prompts the user for the next name.

Operator response

Correct the HOSTS.LOCAL data set and rebuild the HOSTS.ADDRINFO and *user_id.v3r1.*HOSTS.SITEINFO data sets with MAKESITE, as necessary.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

DoIt

EZA0594I **No net named *net***

Explanation

There are no NET entries for the specified name in the HOSTS.ADDRINFO and *user_id.*HOSTS.SITEINFO data sets.

System action

TCPIP continues and prompts the user for the next name.

Operator response

Correct the HOSTS.LOCAL data set and rebuild the HOSTS.ADDRINFO and *user_id.*HOSTS.SITEINFO data sets with MAKESITE, as necessary.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

DoIt

EZA0595I	**MVS TCP/IP Testsite.**

Explanation

Indicates that TESTSITE has been invoked. This is the first message displayed after issuing the TESTSITE command.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

TestSiteInfo

EZA0596I	**Testsite checks site names against HOST.SITEINFO**
EZA0597I	**Prompting for each name until told ‘quit’ For each**
EZA0598I	**name, Testsite describes the corresponding internet**
EZA0599I	**type (HOST, GATEWAY, NET) and internet addresses.**

Explanation

This group of messages is displayed after each site name you enter has been verified against the names found in HOSTS.SITEINFO. These messages are continually displayed until you issue a QUIT command.

System action

TESTSITE continues. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

TestSiteInfo

EZA0600I **To quit, type ‘quit’.**

Explanation

The Testsite utility received a response to its prompt for a site name (see messages EZA0596I - EZA0599I) that was not valid. The only other valid response, other than a site name, is ‘quit’.

System action

TCPIP continues.

Operator response

If you need to quit, enter QUIT and press ENTER.

System programmer response

No action is needed.

Module

CMTSITE

Procedure name

TestSiteInfo

EZA0601E **Error obtaining Userid.**

Explanation

The user ID entered was not found.

System action

TCPIP is unsuccessful.

Operator response

Reissue the user ID, make sure it is correct.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0602I **TCP Host Name is: *MVSA.TCP.RALEIGH.IBM.COM***

Explanation

This message indicates the name of the TCP host.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0604E **Cannot use TCP/IP: *errmsg (msgnum)***

Explanation

TCPIP is not available because of the indicated reason.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP halts.

Operator response

Determine the nature of the error, correct it, and try again.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

HOMETEST

Procedure name

Test

EZA0605I **Using name server to resolve: *MVSA.TCP.RALEIGH.IBM.COM***

Explanation

This message indicates that the program is using a blank.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0609E **Error: Found no IP addresses corresponding to TCP Host Name: *name***

Explanation

The indicated TCP host name could not be resolved because there are no corresponding IP addresses.

System action

The program continues.

Operator response

Correct the HOSTNAME variable in the TCPIP.DATA data set or the records in the name server or sites tables, and rerun HOMETEST. See z/OS Communications Server: IP Configuration Guide for more information on TCPIP.DATA data set.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0610E ***name***

Explanation

This message appears with message EZA0609E and indicates that no addresses were found for the entered host name.

System action

TCPIP is unsuccessful.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0611I **The following IP addresses correspond to TCP Host Name: *name***

Explanation

This message gives a list of Internet Protocol addresses that correspond to the TCP Host Name. These addresses are checked against the HOME list and defined in the *hlq*.PROFILE.TCPIP data set Host Name.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0612I ***list***

Explanation

This message displays with EZA0611I producing a list of the IP addresses that correspond to the TCP Host Name.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0614I **The following IP addresses are the HOME IP addresses defined in the *hlq*.PROFILE.TCPIP data set.**

Explanation

The internet protocol addresses corresponding to your site host name are checked against the HOME list and defined in the *hlq*.PROFILE.TCPIP data set. A warning message is issued if any addresses are missing from the HOME list.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HomeTest

Procedure name

Test

EZA0615I ***list***

Explanation

This message displays with EZA0614I providing a list of the HOME addresses defined in the *hlq*.PROFILE.TCPIP data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0617E **Error: *address* is not in the HOME List**

Explanation

The indicated IP address could not be found in the HOME list in *hlq*.PROFILE.TCPIP. data set.

System action

The program continues.

Operator response

Correct the IP address in the HOME list in the *hlq*.PROFILE.TCPIP data set, the name server, or the site tables, and rerun HOMETEST. See z/OS Communications Server: IP Configuration Guide for more information on *hlq*.PROFILE.TCPIP data set.

Module

HOMETEST

Procedure name

Test

EZA0618I **All IP addresses for *name* are in the HOME list!**

Explanation

This message indicates that all the internet protocol addresses were found in the HOME list.

System action

TCPIP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0619I **Running IBM MVS TCP/IP *release* TCP/IP Configuration Tester**

Explanation

This message indicates the version and level of the data set being tested on the TCPIP system with the HOMETEST statement.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

HomeTest

EZA0621I **The FTP configuration parameter file used will be *data set*.**

Explanation

The initial FTP configuration parameters used during operation of TCPIP are from the indicated source data set. This can be the SYSFTPD DD statement, the configuration data set, or an error message.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0622I **Hometest was successful - all Tests Passed!**

Explanation

All of the configuration data sets have passed the configuration tests.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0623W **Warning: no TCP/IP system parameter file was available for use.**

Explanation

The TCPIP.DATA data set is not found.

System action

HOMETEST ends. Other error messages may occur with this one.

Operator response

Contact the system programmer.

System programmer response

Specify a TCPIP.DATA data set for TCP/IP. For more information, see z/OS Communications Server: IP Configuration Reference.

Module

HOMETEST

Procedure name

Test

EZA0624I *** * * Hometest failed - configuration error * * ***

EZA0625I **Please correct your TCP/IP configuration and run this program again.**

Explanation

These messages appear together to show that there is a configuration error and requests that the error be fixed, and the HOMETEST command reissued.

System action

HOMETEST halts. TCPIP continues.

Operator response

Correct the error in the configuration data set and rerun the program. For more information, see z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

HomeTest

EZA0626E **InitEmulation failed, try again with ECMODE ON**

Explanation

This message is displayed when there is an error in the input data.

System action

TCPIP is unsuccessful.

Operator response

Turn on ECMODE and reissue the HOMETEST command.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0627E **Error: Name resolution via Nameserver was unsuccessful.**

Explanation

The TCP host name could not be resolved using the name server for the indicated reason.

System action

The program continues.

Operator response

Determine why HOMETEST cannot communicate with the name server, and rerun HOMETEST.

System programmer response

No action is needed.

Module

HOMETEST

Procedure name

Test

EZA0628E **Reason: *errmsg (msgnum)***

Explanation

This message indicates that the HOMETEST command was unsuccessful.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP is unsuccessful.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

HOMETEST

Procedure name

Test

EZA0640E **Usage: CONVXLAT InputDsname OutputDsname (KANJI|HANGEUL| TCHINESE -CODEFILE-**

Explanation

Invalid syntax was specified for the CONVXLAT command.

System action

The program halts. No output data set is produced.

Operator response

Reenter the command using valid syntax. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables using CONVXLAT.

System programmer response

No action is needed.

Module

CMCONVXL PASCAL

Procedure name

GetParameters,DoIt

EZA0641E **Input dataset *dataset* not found**

Explanation

The CONVXLAT command specified an input data set that does not exist.

System action

The program halts. No output data set is produced.

Operator response

Reenter the command specifying an existing data set for the InputDsname parameter. Make sure the data set contains valid translation tables. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables using CONVXLAT.

System programmer response

No action is needed.

Module

CMCONVXL PASCAL

Procedure name

DoIt

EZA0642E **Premature EOF on input dataset *dataset***

Explanation

The CONVXLAT command specified an SBCS input data set that ended prematurely, before the output translation table was completed.

System action

The program halts. An invalid output data set is produced.

Operator response

Reenter the command specifying a valid data set for the InputDsname parameter. Make sure the data set contains valid SBCS translation tables. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables using CONVXLAT.

System programmer response

No action is needed.

Module

CMCONVXL PASCAL

Procedure name

DoOneTable

EZA0643E **Error in input line: "*input record*"**

Explanation

The CONVXLAT command specified an SBCS input data set that contained a data error in the specified line. The line did not contain valid hexadecimal codes.

System action

The program halts. An invalid output data set is produced.

Operator response

Reenter the command specifying a valid data set for the InputDsname parameter. Make sure the data set contains valid SBCS translation tables. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables using CONVXLAT.

System programmer response

No action is needed.

Module

CMCONVXL PASCAL

Procedure name

DoOneTable

EZA0644E **Input contains wrong number of bytes: "*input record*"**

Explanation

The CONVXLAT command specified an SBCS input data set that contained a data error in the specified line. The line did not contain 16 valid hexadecimal codes.

System action

The program halts. An invalid output data set is produced.

Operator response

Reenter the command specifying a valid data set for the InputDsname parameter. Make sure the data set contains valid SBCS translation tables. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables using CONVXLAT.

System programmer response

No action is needed.

Module

CMCONVXL PASCAL

Procedure name

DoOneTable

EZA0646E **Output data set DSORG=PO. Execution is HALTed**

Explanation

The output file submitted for the CONVXLAT command is a partitioned data set. The CONVXLAT command will not delete a partitioned data set.

System action

The command is not executed. TCPIP continues.

Operator response

Resubmit the CONVXLAT command with a valid output file. For more information about the CONVXLAT command, see z/OS Communications Server: IP User's Guide and Commands.

System programmer response

Assist the user as necessary.

Module

CMCONVXL

Procedure name

DoIt

EZA0647E **Output dataset is invalid. Execution is HALTed**

Explanation

The output data set specified for the CONVXLAT command was incorrect.

System action

The command is not executed. TCPIP continues.

Operator response

Resubmit the CONVXLAT command with a valid output file. For more information about the CONVXLAT command, see z/OS Communications Server: IP User's Guide and Commands.

System programmer response

Assist the user as necessary.

Module

CMCONVXL

Procedure name

DoIt

EZA0649E **Output dataset not found. Cannot allocate PDS.**

Explanation

The output data set for the CONVXLAT command did not exist when the CODEFILE option was used. The CONVXLAT program is unable to allocate the partitioned data set required for generation of the codefiles.

System action

The program halts. No output data set is produced.

Operator response

For more information about the error, contact the system programmer.

System programmer response

Make sure that the output data set for the codefiles is a correctly allocated partitioned data set. See z/OS Communications Server: IP Configuration Reference for more information about allocating a PDS for CONVXLAT codefile generation.

Module

CMCONVXL PASCAL

Procedure name

DoIt

EZA0650W **DBCS code is defined more than once - *code_index*. Continuing ...**

Explanation

The error occurred during processing of DBCS translation data from the input data set. A DBCS code index was defined more than once.

System action

The program continues. The DBCS code defined more than once will have the value first assigned to it in the input data set.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that all DBCS codes in the data set are defined only once. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0651W **DBCS code index - *code_index* - out of order. Continuing ...**

Explanation

The error occurred during processing of DBCS translation data from the input data set. A DBCS code index was found that was numerically less than the previous code index. The output data set may contain unpredictable results and should not be used.

System action

The program continues.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that all DBCS codes in the data set are defined in numerical order from the lowest value to the highest. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0652I **Current code set is - *code_file***

Explanation

This message is displayed when the CODEFILE option of CONVXLAT is used. It indicates which codefile is currently being generated.

System action

The program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMCONVXL PASCAL

Procedure name

DoCodeFiles

EZA0653W **SBCS code is defined more than once - *code_index*. Continuing ...**

Explanation

The error occurred during processing of SBCS translation data from the input data set. An SBCS code index was defined more than once. This SBCS code index will have the value first assigned to it in the input data set.

System action

The program continues.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that all SBCS codes in the data set are defined only once. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0654W **SBCS code index - *code_index* - out of order. Continuing ...**

Explanation

The error occurred during processing of SBCS translation data from the input data set. An SBCS code index was found that was numerically less than the previous code index. The output data set will contain unpredictable results and should not be used.

System action

The program continues.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that all SBCS codes in the data set are defined in numerical order from the lowest value to the highest. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0655E **No SBCS input data in input dataset *dataset***

Explanation

The input data set contained only DBCS code point mappings and did not contain the associated SBCS data. The output data set will not be complete and should not be used.

System action

The program halts.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that the input data set contains appropriate SBCS code point mappings. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0656E **Empty input dataset *dataset***

Explanation

The input data set existed but contained no data. The output data set will be empty.

System action

The program halts.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that the input data set contains the appropriate translation table data. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0657E **Error in first line of translation data: *input_record***

Explanation

The first line of translation data in the input data set was in error. This error may be caused by an incorrect number of translation data columns or by incorrect (for example, too long, too short, not hex data and so on) data in the columns. The first line of data must contain correct and complete data for processing to continue. The output data set will be empty.

System action

The program halts.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that the input data set contains the appropriate translation data in the first line of uncommented data. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0658E **Error in DBCS input line: *input_record***

Explanation

An error was found processing a line of DBCS translation data from the input data set. Incorrect data was found in one of the columns of the input line. The output data set will contain invalid data and should not be used.

System action

The program halts.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that the input data set contains valid translation data in all DBCS data lines. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0659E **Error in SBCS input line: *input_record***

Explanation

An error was found processing a line of SBCS translation data from the input data set. Incorrect data was found in one of the columns of the input line. The output data set will contain invalid data and should not be used.

System action

The program halts.

Operator response

For more information, contact the system programmer.

System programmer response

Make sure that the input data set contains valid translation data in all SBCS data lines. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Procedure name

DoDbcsBinFile, DoCodeFiles

EZA0660E **Output file is not a PS dataset**

Explanation

The output file specified for the CONVXLAT command was not a sequential data set. The files are not translated.

System action

TCPIP continues.

Operator response

Resubmit the CONVXLAT command specifying a sequential data set for the output file.

System programmer response

Assist the user as necessary.

Module

CMCONVXL

Procedure name

GetParameters

EZA0661E **Error in SBCS input data. Incorrect number of code point mappings.**

Explanation

The input data set for the CONVXLAT command did not contain 256 single byte code point mappings. An output data set is produced but is not valid.

System action

The program halts.

Operator response

For more information about the error, contact the system programmer.

System programmer response

The SBCS translation data must contain exactly 256 code point mappings when using CONVXLAT options KANJI, HANGEUL or TCHINESE. Make sure that the input data set has the correct number of single byte code point mappings. See z/OS Communications Server: IP Configuration Reference for more information about customizing DBCS/SBCS translation tables for use by CONVXLAT.

Module

CMCONVXL PASCAL

Chapter 3. EZA1*xxxx* messages

EZA1450I **IBM FTP *version***

Explanation

This message indicates which version of File Transfer Protocol (FTP) is running on the system.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1451E **Cannot load translate table specified by TRANSLATE parameter *name***

Explanation

The FTP client was started with the TRANSLATE *name* or **-t** *name* parameter on the FTP command. FTP used the name specified to search for a data set containing binary translate tables, but was unable to find a usable data set.

System action

FTP Client exits.

Operator response

Notify the system programmer of the error.

System programmer response

Reissue the FTP command with the TRACE or **-d** parameter to enable tracing which will display the results of each attempt in the search order. Respond to the error indicated.

Module

EZAFTPNX

EZA1456I **Connect to ?**

Explanation

This is a prompt for the user to enter the name or IP address of the host system he wants to connect to. Port number is optional.

System action

Processing continues.

Operator response

Enter host name or IP address.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1457I **You must first issue the 'OPEN' command**

Explanation

The OPEN subcommand must be issued to establish a connection to the other host's FTP server.

System action

FTP continues.

Operator response

Issue the OPEN subcommand with a host name. See the z/OS Communications Server: IP User's Guide and Commands for information about the open subcommand.

System programmer response

No action is needed.

Module

EZAFTPCL, EZAFTPPC

EZA1458I **You must first issue the 'USER' command**

Explanation

The USER subcommand must be issued to identify you to the other host after opening a connection. Your password on the other host can also be specified on the USER subcommand.

System action

FTP continues.

Operator response

Issue the USER subcommand with your logon name on the other host. See the z/OS Communications Server: IP User's Guide and Commands for information about the user subcommand.

System programmer response

No action is needed.

EZA1459I **NAME *host user***

Explanation

This is a prompt for the user ID on the target host system.

System action

None.

Operator response

Type the user ID for the target system or press enter to use the same user ID that you are using on the client system.

System programmer response

No action is needed.

Module

EZAFTPCZ

EZA1460I **Command:**

Explanation

The system is prompting you to enter a command.

System action

The system waits for you to enter a command.

Operator response

Enter the command you want to process.

System programmer response

No action is needed.

Module

EZAFTPCU

EZA1464I **Missing parm after TRANSLATE**

Explanation

A nonstandard translation table name (*data set_name*) must be specified on the FTP command TRANslate parameter. FTP uses the translation table in the *user_id.data set_name.TCPXLBIN* rather than the standard translation table provided with TCPIP for MVS.

System action

FTP continues.

Operator response

Specify the data set name of the translate table you want to take precedence.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1465I **FTP: tcp parameter requires an address space name.**

Explanation

When the TCP parameter is specified on the FTP command, an address space name must be specified.

System action

FTP terminates.

Operator response

Specify an address space name on the TCP parameter of the FTP command.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1466I **FTP: using *name***

Explanation

FTP is using the tcpip stack named *name*. If neither the **-p** nor the (TCP start parameter was used to specify a tcpip stack name, this stack name came from TCPIPJOBNAME in TCPIP.DATA.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1471E **Open of INPUT data set failed**

Explanation

The FTP Client attempted to open the data set allocated to the INPUT or SYSIN DD statement and encountered an error.

System action

The FTP Client terminates.

Operator response

Correct the error with the data set.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1472I **Ambiguous (*command*)**

Explanation

FTP encountered a command during processing that is ambiguous. The command is displayed in the message.

System action

FTP continues.

Operator response

Enter a valid command.

System programmer response

No action is needed.

Module

EZAFTPPC

EZA1473W **LRECL *lrecl* is incompatible with RECFM *recfm***

Explanation

The LRECL value is incompatible with the RECFM. This value might have come from FTP.DATA, as the result of a LOCSITE command, or from MVS as a default.

lrecl is the length of the record.

recfm is the Record format.

System action

The file transfer ends.

Operator response

Enter a correct value using LOCSITE LRECL=value. See the z/OS Communications Server: IP User's Guide and Commands for information about transferring data using the File Transfer Protocol.

System programmer response

Update the FTP.DATA file. See the z/OS Communications Server: IP Configuration Reference and to the z/OS Communications Server: IP User's Guide and Commands for information about the parameters of the FTP.DATA file.

Module

EZAFTPCG

Procedure name

RECVDATA()

EZA1475I **Connection with *foreignhost* terminated**

Explanation

The connection to the foreign host is closed.

System action

FTP Client halts.

Operator response

Reestablish a connection with the foreign host's FTP server by using the OPEN subcommand.

System programmer response

No action is needed.

Module

EZAFTPSC

EZA1478I **GetReply returns *replycode***

Explanation

This message indicates the reply code from the specified host (control connection).

System action

FTP continues.

Operator response

If an error occurred in the connection process, the reply code indicates the error. Correct the error according to the reply code. Otherwise, no action is necessary.

System programmer response

No action is needed.

EZA1479I **File transfer might be incomplete - Last record received without EOL sequence**

Explanation

The user requested a file from the server. The FTP client was receiving the file while ENCODING was set to MBCS. The server sent the file without appending <CRLF>, the FTP end of line (EOL) sequence, to the last record. The FTP client configured MBREQUIRELASTEOL TRUE, so the client considers this to be an error. The file transfer might not be complete.

System action

The FTP client sends an ABOR command to the server. The FTP client continues.

Operator response

No action needed.

System programmer response

If the sender of the file was not a z/OS CS FTP server, consult the vendor of that product for more information. If the vendor indicates that the last record does not include an EOL marker, use the **LOCSIte** subcommand with the NOMBREQUIRELASTEOL parameter set to change the MBREQUIRELASTEOL setting, then get the file again. If the problem continues, contact the vendor.

If the sender of the file was a z/OS CS FTP server, the transfer is incomplete. z/OS FTP appends an EOL marker to the final file record of multibyte files unless you have overridden the sender's EOL marker using the server's MBSENDEOL configuration option. Use the **stat** subcommand to query the server's MBSENDEOL setting. If MBSENDEOL is not CRLF, use the **site** subcommand with the MBSENDEOL=CRLF parameter set to change the server's MBSENDEOL setting, then get the file again. If MBSENDEOL is CRLF, save the output from the **stat** subcommand and the **LOCSTat** subcommand, and contact the IBM support center.

See the information about the MBREQUIRELASTEOL and MBSENDEOL configuration options in z/OS Communications Server: IP Configuration Reference. See the information about using the stat, site, LOCSTat, or LOCSIte subcommands in z/OS Communications Server: IP User's Guide and Commands.

For z/OS FTP products, see the information about support for MBCS languages in z/OS Communications Server: IP User's Guide and Commands for information about configuring z/OS FTP for multibyte transfers. For other FTP implementations, consult your vendor for more information.

User response

Inspect the file to determine if the transfer was complete. If the transfer is not complete, contact the system programmer.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

Not applicable.

Routing code

Not applicable.

Descriptor code

Not applicable.

Example

```
EZA1460I Command:
locsite mbrequirelasteol
EZA1460I Command:
get /tmp/test8.txt 'user1.test8.txt'
EZA1701I >>> PORT 9,42,105,36,4,42
200 Port request OK.
EZA1701I >>> RETR /tmp/trash/test8.txt
125 Sending data set /tmp/trash/test8.txt
EZA1490I Error writing to data set
EZA1479I File transfer might be incomplete - Last record received without EOL
sequence
>>>ABOR
426 Connection closed; transfer aborted.
226 Abort successful.
EZA2108I Confidence=Low for GET of USER1.TEST8.TXT
EZA1460I Command:
```

EZA1480I Invalid argument string

Explanation

The program detected a single quotation mark inside the argument string. This is not valid.

System action

FTP continues.

Operator response

Correct the argument string by removing the single quotation mark from within the string.

System programmer response

No action is needed.

EZA1481I Cannot find closing quote in command string

Explanation

The program cannot find the quotation mark at the end of the command string.

System action

FTP continues

Operator response

Correct the command string by adding a single quotation mark at the end of the command string.

System programmer response

No action is needed.

Module

EZAFTPCP, EZAFTPCG

EZA1485I ***number* bytes transferred - *interval* second interval rate *num_transferred_interval* KB/sec - Overall transfer rate *num_transferred_overall* KB/sec**

Explanation

This message indicates the total number of bytes of binary data transferred to or from the FTP client along with the current rate of transfer.

number is the total number of bytes transferred since the transfer started.

interval is the length of the measured transfer interval in seconds. This rate is set using LOCSITE PROGRESS or the PROGRESS statement in the FTP.DATA file.

num_transferred_interval is the number of KBs (KB equals 1000 bytes) transferred per second in this interval.

num_transferred_overall is the number of KBs (KB equals 1000 bytes) transferred per second since the transfer started.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA1489I **Cannot allocate buffer space**

Explanation

The FTP client attempted to allocate a work buffer and failed. Buffer requirements are moderate so this is unlikely.

System action

The program ends abnormally.

Operator response

Increase REGION below the line to 2M.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1490I **Error writing to data set**

Explanation

Data could not be written to the data set. FTP frees the buffer space.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA1491I **Cannot find end of record character in data stream.**

Explanation

FTP cannot find an end of record character in the data it is transmitting. FTP issues this message and message EZA1492I and frees the data buffer.

System action

FTP continues.

Operator response

See the response for message EZA1492I.

System programmer response

See the response for message EZA1492I.

EZA1492I **Maximum record length supported in EBCDIC/ASCII mode is 64K**

Explanation

FTP could not find an end of record character in the data it is transmitting. The maximum length for an EBCDIC or ASCII mode transmission is 64 KB. FTP does not write the whole buffer and moves any remaining data to the beginning of the buffer. FTP starts getting data at the beginning of the free space in the buffer.

System action

FTP continues.

Operator response

Make sure the appropriate end of record character is in the data to be transmitted.

System programmer response

No action is needed.

EZA1532I **User exit *exitname* module *modname* modified the FTP command arguments**

Explanation

The FTP client displays this message when an FTP client user exit returns a return code of 4. A return code of 4 indicates that the user exit has modified one or more arguments of the command that the client is about to send to the server.

In the message text:

exitname
The name of the FTP client user exit that is associated with *modname*

modname
The name of the FTP client user exit routine that modified the FTP command arguments

System action

The FTP command arguments are modified. More than one exit routine might be associated with this user exit. The FTP client passes the modified arguments to the next user exit routine.

Operator response

Not applicable.

System programmer response

Use the FTP client output to determine whether the modified arguments are acceptable.

- If the modified command is acceptable, no further action is needed.
- Otherwise, correct the user exit routine *modname*.

For more information about the FTP client user exits, see FTP client user exits statement in z/OS Communications Server: IP Configuration Reference.

User response

If you expected the user exit to modify the command arguments, no action is needed. Otherwise, save the FTP client output message for problem determination and contact the system programmer.

Problem determination

The user must enable trace PAR for detailed information about the modified parameters that are passed to the user exit.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data
connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS312.tcp.raleigh.ibm.com, 03:22:10 on 2011-07-05.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1460I Command:
ls /etc/ftp.data
EZA1701I>>> PORT 9,42,105,183,4,4
200 Port request OK.
EZA1532I User exit EZAFCCMD module EZAFCCMD modified the FTP command arguments
EZA1701I >>>NLST /tmp/ftp.data
125 List started OK
EZA2284I /tmp/ftp.data
250 List completed successfully.
EZA1460I Command:
quit
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
#
```

EZA1533I **User exit *exitname* module *modname* prevented user *userid* from sending command *command* – exit reason code x'*hexadecimal-rsncode*' (*decimal-rsncode*)**

Explanation

The FTP client displays this message when an FTP client user exit returns a return code of 8 to reject an FTP command.

In the message text:

exitname
The name of the FTP client user exit that is associated with *modname*

modname
The name of the FTP client user exit routine that rejects the FTP command

userid
The FTP client user ID

command
The FTP command which would have been sent to the FTP server if *modname* had not rejected it

hexadecimal-rsncode
The hexadecimal value of the reason code from the FTP client user exit routine

decimal-rsncode
The decimal value of the reason code from the FTP client user exit routine

System action

The command *command* is not sent to the server. If more than one exit routine is associated with *exitname*, the FTP client stops calling exit routines for *command*.

Operator response

Not applicable.

System programmer response

- Inspect the optional message “EZA1556I” on page 76 and message EZA1533I that the FTP client displays to determine why *modname* rejected the FTP command. If it is appropriate for the exit routine to reject the FTP command, no further action is needed.
- Otherwise, correct the user exit routine.

For more information about the FTP client user exits, see FTP client user exits statement in z/OS Communications Server: IP Configuration Reference.

User response

If you expected the user exit to reject the command, no action is needed. Otherwise, save the message for problem determination and contact the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data
connection.
EZA1450I IBM FTP CS V2R1
```

```
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS312.tcp.raleigh.ibm.com, 03:22:10 on 2011-07-05.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1460I Command:
pwd
EZA1556I EZAFCCMD message: You have no permission to issue PWD command
EZA1533I User exit EZAFCCMD module EZAFCCMD prevented user USER1 from sending command PWD – exit
reason code  x'FFFFFF'  (-1)
EZA1460I Command:
quit
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
```

EZA1534I ***** Control connection with *ForeignHost* dies.**

Explanation

The connection to the other host no longer exists.

System action

The local host waits for the other host to close the connection and then closes the local host end of the connection.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPSC

EZA1542I **Usage: *usage***

Explanation

This is a brief description of the usage of an FTP subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1543I **Please type a longer prefix of the command name**

Explanation

The command prefix you entered is not complete enough to distinguish it from other commands.

System action

The system waits for your response.

Operator response

Enter more characters of the command and resubmit the request.

System programmer response

No action is needed.

Module

EZAFTPPC

EZA1544I **TSO *Command***

Explanation

This message indicates the TSO command that is being processed by FTP.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1545I **Return code *rc* from user exit *exitname* module *modname* is ignored**

Explanation

The FTP client displays this message when it ignores a return code from an FTP client user exit routine.

In the message text:

rc
The return code from the FTP client user exit routine

exitname
The name of the FTP client user exit

modname
The name of the FTP client user exit routine

For more information about why a user exit ignores return codes from an FTP client user exit routine, see FTP client user exits statement in z/OS Communications Server: IP Configuration Reference.

For more information about user exit routines, see using dynamic exits services in z/OS MVS Programming: Authorized Assembler Services Guide.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

No action is needed.

User response

No action is needed.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data
connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS112, 07:40:10 on 2011-06-10.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
```

```
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1460I Command:
quit
EZA1545I Return code 12 from user exit EZAFCCMD module EZAFCCME is ignored
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
EZA1545I Return code 12 from user exit EZAFCREP module EZAFCREE is ignored
```

EZA1546I **User exit *exitname* module *modname* ended the FTP client – exit reason code x'*hexadecimal-rsncode*' (*decimal-rsncode*)**

Explanation

The FTP client displays this message when an FTP client user exit returns a return code of 12 and ends the FTP client.

In the message text:

exitname
The name of the FTP client user exit that is associated with *modname*

modname
The name of the FTP client user exit routine that ends the FTP client

hexadecimal-rsncode
The hexadecimal reason code from the FTP client user exit routine

decimal-rsncode
The decimal reason code from the FTP client user exit routine

System action

The FTP client ends. One or more exit routines might be associated with this user exit.

Operator response

Not applicable.

System programmer response

Inspect the messages the FTP client displays to determine why *modname* ended the FTP client.

- If the explanation provided by message EZA1546I and message "EZA1556I" on page 76 is satisfactory, and it is appropriate for the exit routine to end the FTP client, no further action is needed.
- Otherwise, correct the user exit routine.

  For more information about the FTP client user exits, see FTP client user exits statement in z/OS Communications Server: IP Configuration Reference.

User response

Save the message for problem determination and contact the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data
connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS112, 10:08:45 on 2011-06-10.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1460I Command:
pwd
EZA1556I EZAFCCMD message: Greetings from Raleigh as set by user exit EZAFCCMD
EZA1546I User exit EZAFCCMD module EZAFCCME ended the FTP client – exit reason code x'FFFFFF'
(-1)
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
```

EZA1548I **User exit *exitname* module *modname* abended with abend code x'*rc*' and abend reason code x'*rsncode*'**

Explanation

The FTP client displays this message when an FTP client user exit ends abnormally.

In the message text:

exitname
The name of the FTP client user exit

modname
The name of the FTP client user exit routine

rc
The hexadecimal abend code from the FTP client user exit routine

rsncode
The hexadecimal abend reason code from the FTP client user exit routine

For more information about abend codes and abend reason codes, see the z/OS MVS System Codes.

System action

Dynamic exits services disables this user exit routine. This user exit routine will not be called again until the system programmer takes corrective action.

Operator response

Not applicable.

System programmer response

See the z/OS MVS System Codes for an explanation and a programmer response.

User response

Save the message and the system log for problem determination, and contact the system programmer.

Problem determination

Examine the MVS console or the system log for information about the abend.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS112, 09:36:17 on 2011-06-10.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is
"USER1.".
EZA1460I Command:
pwd
```

```
EZA1701I >>> PWD
257 "'USER1.'" is working directory.
EZA1548I User Exit EZAFCREP Module EZAFCREE abended with abend code x'0C1000' and abend reason code
x'00000001'
EZA1460I Command:
quit
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
EZA1555I CSVDYNEX CALL failed for user exit EZAFCREP, RETURN CODE: x'04' REASON CODE: x'00000406'
```

EZA1549I User exit *exitname* module *modname* returned an invalid return code *rc*

Explanation

The FTP client displays this message when an FTP client user exit routine returns a value in register 15 that is not supported for user exit *exitname*.

In the message text:

exitname
The name of the FTP client user exit

modname
The name of the FTP client user exit routine

rc
The return code from the FTP client user exit routine

System action

The FTP client ends. One or more exit routines might be associated with this user exit. The FTP client stops calling user exit routines when a user exit routine returns an invalid return code.

Operator response

Not applicable.

System programmer response

Inspect the messages the FTP client displays and modify the user exit routine. For more information about user exit routines, see Diagnosing FTP client user exit problems in z/OS Communications Server: IP Diagnosis Guide.

User response

Save the messages for problem determination and contact the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data
connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS112, 11:08:45 on 2011-06-10.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1460I Command:
pwd
EZA1549I User Exit EZAFCCMD module EZAFCCME returned an invalid return code 3
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
```

EZA1550E **You are already connected to *host***

Explanation

You are already connected to the host name you specified, *host*.

System action

Open command processing terminates.

Operator response

Quit or close the current FTP session before attempting to begin a new session.

System programmer response

No action is needed.

Module

EZAFTPCZ

EZA1551I **Unknown host: *host***

Explanation

The host name you have requested a connection to is not a valid host name or is not known to TCPIP.

System action

TCPIP allows you to specify another host name and will attempt to establish a connection using that host name.

Operator response

There are several possible causes. The name may have been entered incorrectly. The name server may not be responding. The resolver may not be querying the correct name server.

System programmer response

Ensure that the host name is accessible via a name server, or defined in HOSTS.SITEINFO and HOSTS.ADDRINFO. See the z/OS Communications Server: IP Configuration Guide for information on domain name systems or HOSTS.SITEINFO and HOSTS.ADDRINFO data sets.

Module

EZAFTPSC

EZA1552I **Nonnumeric port designation: *portname***

Explanation

The port was specified with the nonnumeric value indicated in this message.

System action

TCPIP allows you to specify another port name and will attempt to establish a connection using that port name.

Operator response

Specify a numeric port name. Valid numeric port names are 16 bits long. The well-known port number for FTP is 21; this is the default.

System programmer response

No action is needed.

EZA1553E **Invalid port number. Open fails.**

Explanation

The Open command cannot continue because a specified port number is less than 0 or too large. Port numbers can be a maximum of 16 bits long. The specified number, indicated in this message, exceeds this maximum.

System action

Open command processing terminates.

Operator response

Specify a port name with a maximum length of 16 bits.

System programmer response

No action is needed.

Module

EZAFTPCZ

EZA1554I **FTP Connecting to: *name name* port *number***

Explanation

FTP client is attempting to connect to specified host system.

System action

No action is needed.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPSC

EZA1555I **CSVDYNEX *request* failed for user exit *exitname*, RETURN CODE: *x'rc'* REASON CODE: *x'rsncode'***

Explanation

FTP called the dynamic exits services CSVDYNEX macro on behalf of an FTP client user exit routine, but the CSVDYNEX macro failed.

In the message text:

request
The text string of the CSVDYNEX macro request

exitname
The name of the FTP client user exit

rc
The hexadecimal return code from the CSVDYNEX macro

rsncode
The hexadecimal reason code from the CSVDYNEX macro

For return and reason codes for the CSVDYNEX macro, see *CSVDYNEX - Provide dynamic exit services* in z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

Examine the *rc* and *rsncode* values from the CSVDYNEX macro in *CSVDYNEX - Provide dynamic exit services* in z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN.

- If *rc* and *rsncode* indicate that dynamic exits services disabled the exit routine, a user exit might have abnormally ended. Inspect the system log to determine why the user exit ended abnormally.
- Otherwise, save the client messages and the system log, and contact the IBM service center.

User response

Save the message for problem determination and contact the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data
connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS112, 07:44:17 on 2011-06-10.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1460I Command:
pwd
EZA1701I >>> PWD
257 "'USER1.'" is working directory.
EZA1548I User Exit EZAFCREP Module EZAFCREE abended with abend code x'0C1000' and abend reason code
x'00000001'
EZA1460I Command:
quit
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
EZA1555I CSVDYNEX CALL failed for user exit EZAFCREP, RETURN CODE: x'04' REASON CODE: x'00000406'
```

EZA1556I ***exitname* message: *buffer contents***

Explanation

The FTP client displays this message when any of the following events occurs:

- The FTP client user exit EZAFCCMD returns a return code of 8 and rejects a command.
- The FTP client user exit EZAFCCMD returns a return code of 12 and ends the FTP client.
- The FTP client user exit EZAFCREP returns a return code of 12 and ends the FTP client.

In the message text:

exitname
The name of the FTP client user exit

buffer contents
The 71 bytes of message text provided by the FTP client user exit

System action

When the FTP client user exit EZAFCCMD returns a return code of 8 and rejects a command, the FTP client continues processing.

When the FTP client user exit EZAFCCMD or EZAFCREP returns a return code of 12, the FTP client ends. One or more exit routines might be associated with this user exit. The FTP client stops calling the FTP user exit routines when one exit routine in the calling sequence returns a return code of 12.

Operator response

Not applicable.

System programmer response

This message is followed by message "EZA1546I" on page 69 when a user exit ends the FTP client or by message "EZA1533I" on page 64 when EZAFCCMD rejects a command. See message EZA1546I or EZA1533I for more information.

- If the explanation provided by message EZA1546I or EZA1533I and EZA1556I is satisfactory, and it is appropriate for the exit routine to reject the command or end the FTP client, no further action is needed.
- Otherwise, correct the user exit routine.

  See FTP client user exits statement in z/OS Communications Server: IP Configuration Reference for more information.

User response

Save the message for problem determination and contact the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCU

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
# ftp localhost
EZY2640I Using /etc/ftp.data for local site configuration parameters.
EZYFT25I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the control
connection.
EZYFT31I Using //'CS390.STANDARD.TCPXLBIN' for FTP translation tables for the data connection.
EZA1450I IBM FTP CS V2R1
EZA1466I FTP: using TCPCS
EZA1554I Connecting to localhost 127.0.0.1. port 21
220-FTPD1 IBM FTP CS V2R1 at MVS112, 10:53:45 on 2011-06-11.
220 Connection will close if idle for more than 5 minutes.
EZA1459I NAME (localhost:USER1):
user1
EZA1701I >>> USER user1
331 Send password please.
EZA1789I PASSWORD:

EZA1701I >>> PASS
230 USER1 is logged on.  Working directory is "USER1.".
EZA1556I EZAFCREP message: Greetings from Raleigh as set by user exit EZAFCREP
EZA1546I User exit EZAFCREP module EZAFCREE ended the FTP client  – exit reason code  x'FFFFFF'
(-1)
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
```

EZA1559I Internal debug options disabled

Explanation

Internal debug options are not set on. The flag, DebugFlag, has been set so that no debugging options are active and no extra debugging commands are allowed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1560I Internal debug options enabled

Explanation

The internal debug options are set on. The flag, DebugFlag, has been set so that debugging options are active and extra debugging commands are allowed. Additionally, tracing is active.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1567I **Unknown command: *name***

Explanation

FTP does not recognize the command specified.

System action

FTP continues.

Operator response

Request help for a valid command. z/OS Communications Server: IP User's Guide and Commands contains the valid FTP commands, their abbreviations, and their descriptions.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1568I **You must first connect to some foreign server**

Explanation

You must be connected to a remote host before using this HELP command.

System action

FTP continues.

Operator response

Establish a connection to a remote host by issuing the FTP command. z/OS Communications Server: IP User's Guide and Commands describes the FTP command.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1569I **{} encloses optional parameters; | separates choices:**

Explanation

In the response information for the HELP ALL command, braces enclose optional parameters, and a vertical bar separates choices in the command syntax.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1570I ***use purpose***

Explanation

This is a more detailed description of the use of an FTP subcommand. This is the subcommand syntax, usage, and explanation of the subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1571I **Usage:** ***value***

Explanation

The specified FTP subcommand is no longer valid. The message also tells which subcommand to use in its place.

System action

FTP continues.

Operator response

Specify the indicated subcommand.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1572I ***purpose***

Explanation

This message is issued in response to an incorrectly specified FTP subcommand. It indicates how to get more information about the indicated subcommand.

System action

FTP continues.

Operator response

Specify the indicated HELP command for syntax information about the subcommand you issued.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1573I *** This command can only be used when 'debug' has been turned on**

Explanation

The command for which you have requested help may only be used when the debug facility is active.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1574I **Usage : *value***

Explanation

This is a brief description of the usage of an FTP subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1575I ***purpose***

Explanation

This message provides a more detailed description of the use of an FTP subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1576I **User-FTP understands these commands**

Explanation

This message is concatenated to message EZA1577I which lists the valid FTP commands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1577I ***command***

Explanation

This message indicates the FTP commands that are valid for use in the FTP environment.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1578I **Specify a command by any unambiguous prefix**

Module

EZAFTPCL

EZA1579I **Specify a local data set by qualifier.qualifier... with optional**

Module

EZAFTPCL

EZA1580I **member as (*member*). Enclose fully qualified names in quotes**

Module

EZAFTPCL

EZA1581I **For information about a particular command, say 'HELP command'**

Explanation

These messages provide information about how to specify FTP commands and how to request help for commands using the HELP command. These are issued in response to a HELP request.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1582I **The foreign server has this help:**

Explanation

This message precedes the help information from the remote host. FTP receives the help information from the remote host and displays it to you.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1583I **Note: no password was requested**

Explanation

The remote host does not require a password to log on.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1584I **Note: no account was requested**

Explanation

The remote host does not require account information before you can access files on it. Some hosts do require account information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1585I **leaving dologin**

Explanation

The DoLogin routine has completed processing. This message is issued when tracing is turned on.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1598I **Maximum record length supported is 64K**

Explanation

The character string being transmitted in binary format exceeds the maximum of 65 535 bytes. FTP sets a return code to indicate that the transfer was not successful.

System action

FTP continues.

Operator response

Ensure that the maximum record length of the data you are transmitting is no more than 65 535 bytes and attempt the transfer again.

System programmer response

No action is needed.

EZA1599I **File delimiter is *value***

Explanation

The indicated value is the file delimiter in use by the FTP client. If the current working directory is a z/OS UNIX file system directory, the delimiter is a forward slash (/); otherwise, the delimiter is a period (.).

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1600I **Trace: *trace*, Send Port: *port***

Explanation

This message is issued in response to the LOCSTAT subcommand. The trace value indicates whether the TRACE setting is true or false. True indicates that tracing is set on. The port value indicates whether the SENDPORT setting is true or false. True indicates that automatic sending of the PORT command will take place.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA1601I **Send Site with Put command: *value***

Explanation

This message is issued in response to the LOCSTAT subcommand. The value specified indicates whether the SENDSITE value is set to true or false. True indicates that FTP will send a SITE command that contains record format information for the file or data set when you issue the PUT or MPUT subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA1602I **Connected to:*host*, Port: *port*, logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. The host is the remote host to which you are connected. The port is the port number of the remote host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA1603I **Connected to:*host*, Port: *port*, not logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. The host is the remote host to which you are connected. The port is the port number of the remote host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA1605I **Local Port: *port***

Explanation

This message is issued in response to the LOCSTAT subcommand. The indicated port is the port number of the local host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA1606I **Data type: *format*, Transfer mode: *mode*, Structure: *structure***

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates the FTP data type (ASCII, EBCDIC, image or DBCS), the transfer mode (stream, block, or compressed), and the structure (file or record).

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA1609I **Usage of PORT command is OFF**

Explanation

The PORT command has been toggled off. The SENDPORT subcommand is used to toggle the PORT command off and on. FTP does not send PORT commands for data transfer when you disable PORT commands by toggling off the function.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1610I **Usage of PORT command is ON**

Explanation

The PORT command has been toggled on. The SENDPORT subcommand is used to toggle the PORT command off and on. By default, the SENDPORT subcommand is turned on when you start the system. Each time you use the SENDPORT subcommand, the PORT command is turned alternately on and off. FTP uses a PORT command, by default, when establishing a connection for each data transfer. FTP does not send PORT commands for data transfer when you disable PORT commands by toggling off the function.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1611I **Usage of SITE command with PUT is OFF**

Explanation

The SENDSITE command setting is false, or toggled off. When SENDSITE is off, the automatic sending of SITE commands is off. This means that when you issue the PUT command, FTP will not automatically send a SITE command containing record format information for the file or data set.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1612I **Usage of SITE command with PUT is ON**

Explanation

The SENDSITE command setting is true, or toggled on. When SENDSITE is on, the automatic sending of SITE commands is on. This means that when you issue the PUT command, FTP automatically sends a SITE command containing record format information for the file or data set.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1613I **Internal debug tracing disabled**

Explanation

Debug tracing has been turned off. When the DEBUG subcommand is issued, debug tracing is toggled on or off.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1614I **Internal debug tracing enabled**

Explanation

Debug tracing has been turned on. When the DEBUG subcommand is on, FTP displays each command that is sent to the remote host and the response that is received from the remote host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1615I **Debug tracing not available**

Explanation

The FTP debug tracing facility is not available. The DEBUG option is off. DEBUG is off by default, unless the TRACE parameter was specified on the FTP command.

System action

FTP continues.

Operator response

Issue the DEBUG subcommand to toggle on the debug option.

System programmer response

No action is needed.

EZA1616I **Aborting connection**

Explanation

The data transfer is being ended. The data connection is closing. An FTP reply code 226 is returned indicating that the data connection is closing, and the requested file action was successful. If an FTP reply code 426 is also returned, it indicates that the connection is closed, and that the transfer ended abnormally and was not successful.

System action

FTP continues.

Operator response

If the data transfer was not successful, reestablish the control connection and attempt the data transfer again.

System programmer response

No action is needed.

EZA1617I ***number* bytes transferred in *number* seconds. Transfer rate *value* Kbytes/sec.**

Explanation

This message indicates the number of bytes transferred and the rate of transfer for a data transfer to or from the other host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1618I **Unknown command '*command*'**

Explanation

The command you entered is not known to FTP. This message is concatenated to message EZA1619I.

System action

FTP continues.

Operator response

Enter a valid FTP command or subcommand. See message EZA1619I for more information. See z/OS Communications Server: IP User's Guide and Commands for the syntax and descriptions of FTP commands and subcommands.

System programmer response

No action is needed.

Module

EZAFTPPC

EZA1619I **For a list of the available commands, say HELP**

Explanation

This message instructs you to use the FTP HELP facility to get information about valid FTP commands.

System action

FTP continues.

Operator response

Use the FTP HELP facility by entering HELP, to find information about the command or subcommand you want to use. Enter a valid FTP command or subcommand. See z/OS Communications Server: IP User's Guide and Commands for the syntax and descriptions of FTP commands and subcommands.

System programmer response

No action is needed.

Module

EZAFTPPC

EZA1624W **Local file *file* not found**

Explanation

The FTP client was unable to locate the indicated local data set. This indicates that either the data set name was incorrect or the data set no longer exists.

System action

FTP continues.

Operator response

Check the syntax and resubmit the last command. If the data set was specified correctly, use the LS command to list the local data sets to make sure the data set exists. For more information about the LS command, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA1625I **Store unique is OFF**

Explanation

The SUNIQUE subcommand is off. When SUNIQUE is off, FTP uses a store command (STOR) with the PUT and MPUT subcommands to store files on the remote host. When STOR is used, the remote host will overwrite an existing data set or file, if you try to store a data set or file with the same name.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1626I **Store unique is ON**

Explanation

The SUNIQUE subcommand is on. When SUNIQUE is on, FTP uses a store unique command (STOU) with the PUT and MPUT subcommands and prevents you from overwriting or erasing the existing data set or file on the remote host if you attempt to store a data set or file with the same name. The created foreign data set or file is stored with a unique name. FTP sends the name of the created remote data set or file to the local host, where the data set or file name is displayed on your terminal.

System action

FTP continues.

Operator response

Make a note of the unique name that FTP returns to your terminal for the created data set or file.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1636I ***** I can't open a data-transfer connection:**

Explanation

FTP cannot establish a data connection. The system displays the specific condition that prevented the connection from being established. FTP closes the socket that was acquired for this connection.

System action

FTP will exit, if the exit parameter has been specified.

Operator response

Reissue the FTP command to reestablish the data connection.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1638I **Store unique will send name**

Explanation

A name parameter will be included when the store unique command (STOU) is sent to the server.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1639I **Store unique will not send name**

Explanation

A name parameter will not be included when the store unique command (STOU) is sent to the server.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA1643I **Data set *dsname* was not replaced.**

Explanation

The specified data set was not replaced during data transfer. A local file already exists. If you want to replace the local file, specify GET with the (REPLACE option.

System action

FTP continues.

Operator response

If you want to replace the local file with the file that is being transferred, specify the GET command with the (REPLACE option.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA1645I **LRECL *lrecl* is invalid for RECFM *recfm***

Explanation

After processing the LOCSITE LRECL=*lrecl* command, the client 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 that is compatible with the record format. This message should be followed by EZA2791I, indicating the new value of the LRECL.

Operator response

If the new value for the LRECL is acceptable, no action is necessary, otherwise change the RECFM and LRECL to valid values.

System programmer response

Use LOCSITE to set the correct values for RECFM or LRECL. See z/OS DFSMS Using Data Sets for information about selecting valid values for BLOCKSIZE, LRECL, and RECFM.

Module

EZAFTPMK

Procedure name

SiteCheck®

EZA1680W **Member names must be specified with partitioned data sets**

Explanation

A file member was specified incorrectly. File members must be specified using partitioned data sets such as the PROFILE.TCPIP data set. The file set is not accepted.

System action

FTP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1681W **Local file already exists**

Module

EZAFTPCG

EZA1682W **To replace it, use the command with the (REPLACE option**

Explanation

This message indicates that a file to be transferred has the same name as a local file. The file is not transferred.

System action

FTP continues.

Operator response

Rename the local file and resubmit the command, or resubmit the command with the (REPLACE option, as shown in the following example:

```
get filename (Replace
```

System programmer response

No action is needed.

Module

EZAFTPCG

EZA1684W **Local file not found**

Explanation

A local file that was to be transferred was not found. The file is not transferred.

System action

FTP continues.

Operator response

Correct the file name and resubmit the command. If the error persists, check the local directory using the LS command to make sure the file exists. For more information about the LS command, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1685W **Invalid local file identifier**

Explanation

A local file that was to be transferred has an incorrect file identifier. The file is not transferred.

System action

FTP continues.

Operator response

Correct the file identifier and resubmit the command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1686W **Invalid character in local file identifier**

Explanation

A local file that was to be transferred was submitted with an incorrect character in the file identifier. The file is not transferred.

System action

FTP continues.

Operator response

Resubmit the command with the correct file identifier.

System programmer response

No action is needed.

EZA1687W **Failure attempting to store a member of a partitioned data set**

Explanation

FTP was unable to store a member of a partitioned data set because the name was typed incorrectly or because the member no longer exists. The file is not transferred.

System action

FTP continues.

Operator response

Check to make sure that the member is still in storage. If the member is still in storage, resubmit the command with the correct name.

System programmer response

No action is needed.

EZA1690W **Failure attempting to close the local data set**

Explanation

FTP was unable to close the local data set because the data set name was given incorrectly or the data set no longer exists.

System action

FTP continues.

Operator response

Check to make sure the data set exists. If so, resubmit the command with the correct data set name.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA1691I **srestart not supported for structure *structure***

Explanation

An **srestart** subcommand was issued to restart a file transfer, but the structure setting is not **file**.

structure is the current structure setting.

System action

FTP stops processing **srestart** and prompts for another subcommand.

Operator response

If file structure was in effect during the original file transfer, use the structure file subcommand to change the structure to file and issue **srestart** again. If file structure was not in effect during the original file transfer, you cannot use **srestart** to restart the file transfer.

System programmer response

No action is needed.

Module

EBAFTPCB

Procedure name

srestart()

EZA1692I **srestart not supported for mode *mode***

Explanation

An **srestart** subcommand was issued to restart a file transfer, but the current mode is not valid for **srestart**. Only stream mode is valid for **srestart**.

mode is the current mode.

System action

FTP stops processing **srestart** and prompts for another subcommand.

Operator response

If stream mode was in effect during the original file transfer, issue the mode stream subcommand to change the mode to stream and issue **srestart** again. If stream mode was not in effect during the original file transfer, you cannot use **srestart** to restart the file transfer.

System programmer response

No action is needed.

Module

EBAFTPCB

Procedure name

srestart()

EZA1693I **srestart not supported for filetype *type***

Explanation

An **srestart** subcommand was issued to restart a file transfer, but the current file type is not valid for **srestart**. Only filetype=SEQ is valid for **srestart**.

type is the current file type setting.

System action

FTP stops processing **srestart** and prompts for another subcommand.

Operator response

If the SEQ file type was in effect during the original file transfer, issue the site filetype=SEQ subcommand to change the file type to SEQ and issue **srestart** again. If SEQ file type was not in effect during the original file transfer, you cannot use **srestart** to restart the file transfer.

System programmer response

No action is needed.

Module

EZAFTPCB

Procedure name

srestart()

EZA1694I **srestart not supported for option sunique**

Explanation

An **srestart** subcommand was issued to restart a file transfer, but the sunique (store-unique) option is in effect.

System action

FTP stops processing **srestart** and prompts for another subcommand.

Operator response

Use the **get** or **put** subcommands instead of **srestart** to transfer the file.

System programmer response

No action is needed.

Module

EZAFTPCB

Procedure name

srestart()

EZA1695I **srestart not supported for data type *type***

Explanation

An **srestart** subcommand was issued to restart a file transfer, but the current data type is not valid for **srestart**. Valid data types for **srestart** are **I** (binary), **A** (ASCII), and **E** (EBCDIC).

type is the current data transfer type as described in the z/OS Communications Server: IP User's Guide and Commands.

System action

FTP stops processing **srestart** and prompts for another subcommand

Operator response

Set the data type to the data type in effect during the original file transfer, then issue **srestart** again. If the data type in effect during the original file transfer is not valid for **srestart**, you cannot use **srestart** to restart the file transfer. You must use the **get** or **put** subcommands to transfer the file.

System programmer response

No action is needed.

Module

EZAFTPCB

Procedure name

srestart()

EZA1696I **srestart *option* failed - issue *subcommand***

Explanation

An **srestart** subcommand was issued to restart a file transfer, but ftp could not restart file transfer.

option is the option specified on the **srestart** subcommand.

subcommand is the subcommand suggested by FTP to transfer the file.

System action

FTP stops processing **srestart** and prompts for another subcommand.

Operator response

Issue the *subcommand* to transfer the file.

System programmer response

If the FTP trace was active, you can check the FTP trace output for error messages related to **srestart** option. See the following publications for more information about the FTP trace:

- z/OS Communications Server: IP User's Guide and Commands
- z/OS Communications Server: IP Configuration Guide
- z/OS Communications Server: IP Configuration Reference
- z/OS Communications Server: IP Diagnosis Guide

Module

EZAFTPCB

Procedure name

srestart()

EZA1698I **srestart failed - SIZE response is not valid**

Explanation

An **srestart** subcommand was issued to resume a file transfer from the client to the FTP server. The client sent a SIZE command to the server to determine how many bytes had already been transferred. The SIZE response returned by the server is not valid, possibly because it points past the end of the client file. The client could not complete the request.

System action

FTP stops processing the current subcommand and prompts for the next subcommand.

Operator response

Report the error to the system programmer.

System programmer response

Verify that the source and destination files have not changed since the original file transfer was interrupted, and that the **srestart** subcommand was valid for restarting the file transfer. See the z/OS Communications Server: IP User's Guide and Commands for information about the **srestart** subcommand. If the files have not changed, and **srestart** was used correctly, report the error to the administrator of the FTP server.

Module

EZAFTPCP

Procedure name

hfs_sndFile()

EZA1699I **SRESTART { put localfile [*remotefile*] | get remotefile [*localfile*]}**

Explanation

This message displays the correct format for the srestart subcommand, which is used to restart a stream mode file transfer. See the z/OS Communications Server: IP User's Guide and Commands for more information about the **srestart** command.

System action

None.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

usage()

EZA1700I **restart a stream mode file transfer**

Explanation

This message displays the purpose of the **srestart** subcommand. See the z/OS Communications Server: IP User's Guide and Commands for more information about the **srestart** command.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

dohelp()

EZA1701I **>>>**

Explanation

This message precedes echoes such as PASS ******** and ACCT ******** to acknowledge user-submitted data.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA1702W **Transmission mode *mode* not implemented for type B.**

Explanation

The data type B (double Byte) is not supported while in block or compress transmission mode.

System action

FTP continues.

Operator response

Either change to mode s (stream) or change the data type to EBCDIC and reissue the command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1719I ***subcommand* not supported for UNIXFILETYPE=FIFO**

Explanation

A subcommand was issued to restart a file transfer to or from a file in the client z/OS UNIX file system, but the value currently configured for the UNIXFILETYPE parameter is not valid for the subcommand. Only UNIXFILETYPE FILE is valid for a subcommand when the local file is a z/OS UNIX file.

See the information about the UNIXFILETYPE statement in z/OS Communications Server: IP Configuration Reference and the LOCSIte subcommand inz/OS Communications Server: IP User's Guide and Commands for information about the UNIXFILETYPE configuration option.

In the message text:

subcommand
The subcommand that you entered to restart a file transfer.

System action

FTP stops processing the subcommand specified by the *subcommand* value and prompts you to enter another subcommand.

Operator response

Not applicable.

System programmer response

None.

User response

If UNIXFILETYPE FILE was configured during the original file transfer, you can issue the LOCSIte subcommand to change the UNIXFILETYPE value back to FILE and issue the specified subcommand again or you can start the interrupted file transfer from the beginning.

If UNIXFILETYPE FIFO was configured during the original file transfer, you cannot resume the interrupted file transfer with a REStart or SRestart subcommand. See the information about restarting a failed data transfer in z/OS Communications Server: IP User's Guide and Commands.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcb

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA1460I Command:
srestart get /etc/hosts
EZA1719I srestart not supported for UNIXFILETYPE=FIFO
EZA1696I srestart get failed - issue get
EZA1460I Command:
```

EZA1720I REPLACE option not allowed for UNIXFILETYPE=FIFO

Explanation

You entered a **`get`** or **`mget`** subcommand, specifying the REPLACE option and a z/OS UNIX local file when UNIXFILETYPE= FIFO was configured. The REPLACE option is not valid for z/OS UNIX files when UNIXFILETYPE=FIFO is configured. If the local file is a z/OS UNIX named pipe, you cannot overwrite the contents of the named pipe. Writing into a named pipe always appends to the contents of the named pipe.

System action

FTP stops processing the current subcommand and prompts you to enter another subcommand.

Operator response

Not applicable.

System programmer response

None.

User response

If the local file is a z/OS UNIX regular file, issue the LOCSIte subcommand with the UNIXFILETYPE=FILE parameter. Then issue the original subcommand again.

If the local file is a z/OS UNIX named pipe, issue the subcommand without the REPLACE parameter. The incoming data will be appended to the contents of the named pipe.

Problem determination

To determine whether a z/OS UNIX file is a regular file or a named pipe, use the z/OS UNIX **`ls`** command. See the information about the long output format in z/OS UNIX System Services Command Reference.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcg.c

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA1460I Command:
get /etc/hosts /tmp/trash/x (rep
EZA1720I REPLACE option not allowed for UNIXFILETYPE=FIFO
EZA1460I Command:
```

EZA1721W Server not responding, closing connection.

Explanation

The FTP client is closing the connection to the FTP server because the server is not responding.

System action

FTP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the Ping command to make sure that the server is accessible to the system. If the server is accessible, check to make sure it is executing FTP properly. See the z/OS Communications Server: IP System Administrator's Commands for more information about the Ping command.

EZA1722I UNIXFILETYPE=FIFO requires FILETYPE=SEQ

Explanation

You attempted to transfer to or from a file in the z/OS UNIX file system when UNIXFILETYPE=FIFO was configured. The current FILETYPE value is not SEQ. This combination of values is not supported.

System action

FTP stops processing the current subcommand, and prompts you to enter the next subcommand.

Operator response

None.

System programmer response

None.

User response

Do one of the following to set a valid combination of parameters.

- If the local file is a z/OS UNIX regular file, issue the LOCSIte subcommand with the UNIXFILETYPE=FILE parameter to change the UNIXFILETYPE value.
- If the local file is a z/OS UNIX named pipe, issue the LOCSIte subcommand with the FILETYPE=SEQ parameter to change FILETYPE.

Then reissue the file transfer subcommand.

Problem determination

To determine whether a z/OS UNIX file is a regular file or a named pipe, use the z/OS UNIX **`ls`** command. See the information about the long output format in z/OS UNIX System Services Command Reference.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcg.c, ezaftpcp.c

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
Command:
get /etc/hosts /tmp/x
UNIXFILETYPE=FIFO requires FILETYPE=SEQ
Command:
```

EZA1725I ***subcommand* fails -- *localFile* is not a named pipe**

Explanation

The subcommand that you issued specified a file in the client z/OS UNIX file system as the local file. The local file is not a z/OS UNIX named pipe, and UNIXFILETYPE=FIFO is configured. A local file in the z/OS UNIX file system must be a named pipe when UNIXFILETYPE=FIFO is configured.

For more information about the UNIXFILETYPE configuration option, see the following sources:

- The UNIXFILETYPE statement in z/OS Communications Server: IP Configuration Reference
- The LOCSIte subcommand in z/OS Communications Server: IP User's Guide and Commands

- The information about using z/OS UNIX System Services named pipes in z/OS Communications Server: IP User's Guide and Commands

In the message text:

subcommand
The subcommand that failed.

localFile
The local file in the client's z/OS UNIX file system.

System action

FTP stops processing subcommand, and prompts you to enter the next subcommand.

Operator response

None.

System programmer response

None.

User response

If you intended to specify a named pipe, do one of the following:

- Reissue the subcommand with a z/OS UNIX named pipe specified as the local file.
- If the subcommand is **`get`** or **`mget`**, reissue the subcommand and specify a name that does not yet exist in the z/OS UNIX file system.

If you did not intend to specify a named pipe, issue the LOCSIte subcommand with the UNIXFILETYPE=FILE parameter. Then issue the subcommand again.

Problem determination

To determine whether a z/OS UNIX file is a regular file or a named pipe, use the z/OS UNIX **`ls`** command.

To create a named pipe, use the z/OS UNIX **`mkfifo`** or **`mknode`** command.

See the information about the long output format in z/OS UNIX System Services Command Reference.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcg

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA1460I Command:
locsite unixfiletype=fifo
EZA1460I Command:
put /etc/hosts
EZA1725I put fails -- /etc/hosts is not a named pipe
EZA1726I local file must be a named pipe when  UNIXFILETYPE=FIFO
EZA1460I Command:

EZA1460I Command:
get /etc/hosts
EZA1725I Get fails -- /etc/hosts is not a named pipe
EZA1726I local file must be a named pipe when  UNIXFILETYPE=FIFO
EZA1460I Command:
```

EZA1726I local file must be a named pipe when UNIXFILETYPE=FIFO

Explanation

You issued a subcommand that specified a file in the z/OS UNIX file system as the local file. The local file is not a named pipe. The local file must be a named pipe when UNIXFILETYPE=FIFO is configured.

This message is preceded by message EZA1725I, which identifies the subcommand that you issued and the local file that you specified.

System action

The FTP client stops processing the current subcommand, and prompts you to enter the next subcommand.

Operator response

None.

System programmer response

None.

User response

See message EZA1725I for more information.

Problem determination

See message EZA1725I for more information.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcg, ezaftpcp.

Routing code

10

Descriptor code

12

Automation

None.

Example

```
locsite unixfiletype=fifo
EZA1460I Command:
put /etc/hosts
EZA1725I put fails -- /etc/hosts is not a named pipe
EZA1726I local file must be a named pipe when  UNIXFILETYPE=FIFO
EZA1460I Command:
```

EZA1733I **waiting up to *seconds* seconds for *process* process to open *localNamedPipe***

Explanation

You issued a subcommand that specified a z/OS UNIX named pipe as the local file. The FTP client must wait for another process to open the named pipe before it can continue processing.

In the message text:

seconds
The number of seconds that the FTP client will wait for another process to open the named pipe. You can use the FIFOOPENTIME configuration option to change this value.

process
Indicates whether the other process must open the named pipe for writing or reading. Valid values are:

read
FTP expects another process to open the named pipe for reading.

write
FTP expects another process to open the named pipe for writing

localNamedPIpe
The z/OS UNIX named pipe that you specified. See the information about using z/OS UNIX System Services named pipes in z/OS Communications Server: IP User's Guide and Commands.

System action

The FTP client waits up to the number of seconds specified by the *seconds* value for another process to open the named pipe. If another process opens the named pipe as expected, FTP continues processing the current subcommand. If another process does not open the named pipe within the time specified, FTP issues message EZA1734I and stops processing the current subcommand.

Operator response

None.

System programmer response

Start the process that will read from or write to he named pipe.

User response

None.

Problem determination

See the user response.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcg.c, ezaftpcp.c

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

In the first example, the user set the UNIXFILETYPE configuration option to FIFO and the FIFOOPENTIME configuration option to 30. The user issued a PUt subcommand that specified the /tmp/sample.fifo named pipe. FTP waited for another process to open the named pipe for writing. When another process opened the named pipe, FTP was able to continue processing the PUt subcommand.

```
EZA1460I Command:
locsite unixfiletype=fifo fifoopentime=30
EZA1460I Command:
put /tmp/sample.fifo
EZA1733I waiting up to 30 seconds for write process to open /tmp/sample.fifo
EZA1701I >>> PORT 9,42,105,36,4,10
200 Port request OK.
EZA1701I >>> STOR sample.fifo
125 Storing data in the Null directory (*dev.null).
250 Transfer completed successfully.
EZA1617I 787 bytes transferred in 1.260 seconds.  Transfer rate 0.62 Kbytes/sec.
EZA1460I Command:
```

In this example, the user entered a PUt subcommand that specified the /tmp/sample.fifo named pipe. FTP waited for 30 seconds for another process to open the named pipe for writing. No other process opened the named pipe. FTP issued message EZA1734I to indicate that it stopped waiting.

```
EZA1460I Command:
put /tmp/sample.fifo
EZA1733I waiting up to 30 seconds for write process to open /tmp/sample.fifo
EZA1734I timer expired waiting for write process to open /tmp/sample.fifo
EZA1460I Command:
```

EZA1734I **timer expired waiting for *process* process to open *localFile***

Explanation

The FTP client tried to open a named pipe in the z/OS UNIX file system. The client waited for the number of seconds specified by the FIFOOPENTIME configuration option. The FIFOOPENTIME time limit expired before another process opened the named pipe.

In the message text:

process

Indicates whether FTP was waiting for another process to open the named pipe for writing or reading. Valid values are:

read

FTP was trying to open the named pipe for writing as part of inbound file transfer. The FTP client cannot write to a named pipe until another process on the client host opens the named pipe for reading.

write

FTP was trying to open the named pipe for reading as part of outbound file transfer. The FTP client cannot read from a named pipe until another process on the client host opens the named pipe for writing.

localFile

The name of the named pipe.

Message EZA1733I precedes this message, and indicates how long the FTP client waits for another process to open a named pipe.

See information about the UNIXFILETYPE (FTP client and server) statement and the FIFOOPENTIME (FTP client and server) statement in z/OS Communications Server: IP Configuration Reference.

System action

FTP stops processing the current subcommand, and prompts for another subcommand.

Operator response

None.

System programmer response

Start a process at the FTP client host:

- When the *process* value is `write`, start the process that will write to the named pipe.
- When the *process* value is `read`, start the process that will read from the named pipe.

After you start the process, attempt the file transfer again.

User response

Perform the following steps:

1. Consider increasing the FIFOOPENTIME configured value before attempting the file transfer again.
   a. Inspect message EZA1733I to determine the FIFOOPENTIME configured value.
   b. Issue the LOCSIte subcommand with the FIFOOPENTIME parameter to specify a larger value.

   See the information about the LOCSIte subcommand in z/OS Communications Server: IP User's Guide and Commands for information about the FIFOOPENTIME configuration option.
2. Start the file transfer again. If restarting the file transfer does not solve the problem, notify the System Programmer.

Problem determination

None.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcg, ezaftpcp

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA1460I Command:
get /etc/hosts /tmp/sample.fifo
EZA1733I waiting up to 30 seconds for read process to open /tmp/sample.fifo
EZA1734I timer expired waiting for read process to open /tmp/sample.fifo
EZA1460I Command:

put /tmp/sample.fifo
EZA1733I waiting up to 30 seconds for write process to open /tmp/sample.fifo
EZA1734I timer expired waiting for write process to open /tmp/sample.fifo
EZA1460I Command:
```

EZA1735I **Std Return Code = *rc*, Error Code = *ec***

Explanation

FTP has ended with an error only because the user has specified EXIT as a start parameter. Normally FTP would continue.

rc is the standard return code. *rc* is in the format **yyxxx**, where

yy
The subcommand code, which is a number from 0 to 99.

xxx
This is the last reply code received from the FTP server. If a subcommand failed before a reply was received from the server, this value is 000.

ec is the client error code set by the FTP client.

System action

The FTP client ends.

Operator response

To determine the cause of the error, see FTP return codes in z/OS Communications Server: IP User's Guide and Commands for information about FTP return codes, including subcommand codes, client error codes, and reply codes.

System programmer response

No action is needed.

Module

EZAFTPCX

EZA1736I ***Argument***

Explanation

FTP is echoing valid arguments from user-submitted data or indicating a QUIT command was generated by unexpected end of file.

System action

FTP continues when echoing valid arguments or terminates when unexpected end of file is encountered.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCU

Module

EZAFTPCX

EZA1737E **Command input too long.**

Explanation

The FTP command input line exceeds the expected length.

System action

FTP bypasses the command and continues.

Operator response

Correct the input command stream.

System programmer response

No action is needed.

Module

EZAFTPCU

EZA1739I ***subcommand*** **failed - timer expired while reading from named pipe**

Explanation

While the FTP client was sending data from the named pipe to the FTP server, it tried to read from a named pipe in the client z/OS UNIX file system. The client waited for the number of seconds specified by the FIFOIOTIME configured value for the read to complete. One reason for this timeout is that the process that is writing to the named pipe stopped writing data without closing the named pipe.

Any data that was sent from the client to the server before the timer expired has been permanently removed from the named pipe.

See the following sources:

- The information about the FIFOIOTIME (FTP client and server) statement in z/OS Communications Server: IP Configuration Reference
- The information about the LOCSIte subcommand in z/OS Communications Server: IP User's Guide and Commands
- The information about using z/OS UNIX System Services named pipes in z/OS Communications Server: IP User's Guide and Commands

In the message text:

subcommand

The subcommand that failed because FTP could not read from the named pipe.

System action

FTP stops processing the current subcommand, and prompts for the next subcommand.

Operator response

None.

System programmer response

Perform the following steps:

1. Consider increasing the FIFOIOTIME configured value.
   a. Use the LOCSTat subcommand and specify the FIFOIOTIME parameter to query the current FIFOIOTIME value.
   b. Use the LOCSIte subcommand and specify the FIFOIOTIME parameter to increase the FIFOIOTIME value

   If you anticipate long idle periods, consider increasing local timers to prevent connection timeouts. See the information about the DATACTTIME (FTP client) statement, the FTPKEEPALIVE (FTP client and server) statement, and the DATAKEEPALIVE (FTP client and server) statement in z/OS Communications Server: IP Configuration Reference for more information about FTP client timers.
2. Determine why the process that writes to the named pipe stopped writing data, but did not close the named pipe.
3. Reissue the subcommand. Any data that was sent from the client to the server before the timer expired has been removed from the named pipe. You must take this into consideration before repeating the file transfer.

User response

Report the problem to the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcp

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
put /tmp/sample.fifo
EZA1733I waiting up to 60 seconds for write process to open /tmp/sample.fifo
EZA1701I >>> PORT 9,42,105,36,4,36
200 Port request OK.
EZA1701I >>> STOR sample.fifo
125 Storing data set
/u/user1/sample.fifo

EZA1739I put failed -- timer expired while reading from named pipe
250 Transfer completed
successfully.
EZA1460I Command:


EZA1460I Command:
appe /tmp/sample.fifo  /u/user1/sample.fifo
EZA1733I waiting up to 60 seconds for write process to open /tmp/sample.fifo
EZA1701I >>> PORT 9,42,105,36,4,42
200 Port request OK.
EZA1701I >>> APPE /u/user1/sample.fifo
125 Appending to data set /u/user1/sample.fifo
EZA1739I append failed -- timer expired while reading from named pipe
250 Transfer completed successfully.
EZA1460I
Command:
```

EZA1764W **FTP: ExitCode *‘code’* is out of range.**

Explanation

The ExitCode is out of range. The correct range is from 0 to 4095.

System action

FTP halts.

Operator response

Correct the value specified for the EXIT= start parameter.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1767I **Source IP address *address* is not valid**

Explanation

The address specified on the -s parameter of the FTP command is not valid. The address must be a valid IPv4 or IPv6 address and cannot be INADDR_ANY, IN6ADDR_ANY, or an IPv4-mapped IPv6 address.

In the message text:

address
The IP address specified for the -s parameter on the FTP command.

System action

The FTP command fails.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the FTP command specifying a valid source IP address.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCY

Routing code

10

Descriptor code

12

Example

```
EZA1767I Source IP address 0.0.0.0 is not valid
```

EZA1768I **Timeout *'timeout'* out of range. Must be between 15 and 86400. Default used.**

Explanation

The timeout value specified is not in the valid range of 15 - 86400. The default timeout value is used instead.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1770I **Source IP address is missing**

Explanation

The -s parameter was specified on the FTP command but the IP address was omitted.

System action

The FTP command fails.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the FTP command, specifying a valid source IP address. See the information about the FTP command in z/OS Communications Server: IP User's Guide and Commands.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCY

Routing code

10

Descriptor code

12

Example

```
EZA1770I Source IP address is missing
```

EZA1771I ***subcommand* fails: *name* is of an unsupported type**

Explanation

The data set name type (DSNTYPE) or organization (DSORG) of data set *name* is not supported by the *subcommand*. The supported data set types are:

- z/OS physical sequential data sets
- z/OS partitioned data sets
- z/OS libraries
- z/OS generation data sets

In the message text:

subcommand
The name of the FTP subcommand. The value can be MVSGET or MVSPUT.

name
The name of the data set that is specified on the FTP subcommand.

For more information about the MVSGET or MVSPUT subcommand, see the z/OS Communications Server: IP User's Guide and Commands.

System action

MVSGET or MVSPUT subcommand fails. FTP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the subcommand with a data set that is of a supported type.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
mvsget 'user1.ftp.ps'   /tmp/
test
EZA1771I MVSGET fails: /tmp/test is of an unsupported type
```

EZA1772I **FTP: EXIT has been set.**

Explanation

An exit has been set. FTP will exit the session for certain FTP errors with a nonzero MVS return code. See the z/OS Communications Server: IP User's Guide and Commands for information about FTP EXIT Return Codes.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA1773I **Unable to delete *dsname* (Error code: *error_code* Reason code: *reason_code* S99ERSN code: *s99reason_code*)**

Explanation

FTP attempted to delete the data set *dsname*, but the deletion did not complete successfully. Possible causes for this error include:

- The data set is in use by someone else.
- The data set has an unexpired retention period.
- The data set is located on a tape volume.
- The direct access volume containing the data set is not write accessible.

In the message text:

dsname
The name of the data set

error_code
S99ERROR field from DYNALLOC

reason_code
S99INFO field from DYNALLOC

s99reason_code
S99ERNS field from DYNALLOC

For more information about the S99ERROR, S99INFO, and S99ERNS fields from DYNALLOC, see the z/OS MVS Programming: Authorized Assembler Services Guide.

System action

FTP continues processing.

Operator response

No action is needed.

System programmer response

The *error_code*, *reason_code*, and *s99reason_code* values displayed contain the S99ERROR, S99INFO, and S99ERNS fields from DYNALLOC. See Interpreting DYNALLOC return codes in z/OS MVS Programming: Authorized Assembler Services Guide to determine why the data set cannot be deleted and to identify the actions to take to correct the problem.

User response

Save the message for problem determination and contact the system programmer.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
mvsget 'user1.ftp.ps'   'user1.ftp.pds'
(reallocate
EZA1773I Unable to delete 'user1.ftp.pds' (Error code: 0210 Reason code: 0000 S99ERSN code:
X'00004379')
EZA1460I Command:
```

EZA1775I ***dsname* deleted**

Explanation

This message indicated that the data set was deleted.

In the message text:

dsname
The name of the data set that was deleted

System action

The data set is deleted. FTP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

No action is needed.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
mvsget 'user1.ftp.ps'   'user1.ftp.pds'  (REAllocate
EZA1701I >>> XDSS 'user1.ftp.ps'
200-LASTREF=2012/01/19 DSEMPTY=FALSE
200 SITE DSNTYPE=BASIC RECFM=VB BLKSIZE=6233 LRECL=256 PRIMARY=1 SECONDARY=1 TRA
CKS EATTR=SYSTEM
EZZ9815I local site variables have changed
EZA1775I USER1.FTP.PDS deleted
EZA1701I >>> PORT 127,0,0,1,4,2
200 Port request OK.
EZA1701I >>> RETR 'user1.ftp.ps'
125 Sending data set USER1.FTP.PS
250 Transfer completed successfully.
EZA1617I 185 bytes transferred in 0.010 seconds.  Transfer rate 18.50 Kbytes/sec.
```

EZA1776I ***subcommand* fails: the target generation data set must be a positive relative reference**

Explanation

The message is displayed when the FTP *subcommand* is transferring into a generation data set by using a negative or current reference.

In the message text:

subcommand
The subcommand name that is specified on the FTP subcommand. The value can be MVSGET or MVSPUT.

For more information about the MVSGET or MVSPUT subcommand, see the z/OS Communications Server: IP User's Guide and Commands.

For more information about generation data sets, see the z/OS DFSMS Using Data Sets.

System action

FTP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the subcommand by using a positive relative reference as the target generation data set or by not using a generation data set as the target data set.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
mvsget 'user.library'
'user1.gdg.mygdgb(-1)'
EZA1776I MVSGET fails: the target generation data set must be a positive relative reference
EZA1460I Command:
```

EZA1777I ***subcommand* fails: the target generation data set cannot be a library**

Explanation

You issued the FTP *subcommand* to transfer a library into a generation data set, but the target generation data set cannot be a library.

In the message text:

subcommand
: The subcommand name you issued to transfer a library into a generation data set

For more information about the data set organization of generation data sets, see the z/OS DFSMS Using Data Sets.

System action

FTP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the subcommand with a different source or target data set.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
mvsget 'user.library' 'user1.gdg.mygdgb(+1)'
EZA1701I >>> XDSS 'user.library'
200-LASTREF=2011/11/08 DSEMPTY=FALSE
200 SITE PDSTYPE=PDSE RECFM=U BLKSIZE=32760 DIRECTORY=10 LRECL=0 PRIMARY=15 SECON
DARY=1 EATTR=SYSTEM
CYLINDERS
```

```
EZA1777I MVSGET fails: the target generation data set cannot be a library
EZA1460I Command:
```

EZA1778I ***subcommand* fails: the target generation data set must be referenced using its absolute name when the source data set is a PDS**

Explanation

When you are transferring a PDS into a generation data set, you must reference the generation data set by its absolute data set name.

In the message text:

subcommand
The subcommand name you issued specifying a target generation data set

System action

FTP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the subcommand using an absolute data set name as the target generation data set, as shown in the following example:

```
mvsget 'user1.ftp.ps'  'user1.gdg.mygdgb.g0007v00'
```

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
mvsget 'user1.ftp.pds' 'user1.gdg.mygdgb(+1)'
EZA1701I >>> XDSS 'user1.ftp.pds'
200-LASTREF=2011/11/08 DSEMPTY=FALSE
200 SITE PDSTYPE=PDS RECFM=VB BLKSIZE=6233 DIRECTORY=27 LRECL=256 PRIMARY=1 SECO
NDARY=1 EATTR=SYSTEM
TRACKS
EZA1778I MVSGET fails: the target generation data set must be referenced using
its absolute name when the source data set is a PDS
```

EZA1779I *subcommand* fails: the target data set already exists

Explanation

The target data set you specified on the FTP *subcommand* exists, but you did not specify the REALLOCATE option.

This is the first message in a group. See the following complete description of the message group:

```
EZA1779I subcommand fails: the target data set already exists
EZA1780I To reallocate it, issue the subcommand with the REALLOCATE option
```

In the message text:

subcommand
The subcommand you issued specifying an existing data set as the target data set

System action

FTP continues processing. The subcommand fails.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Reissue the subcommand with a non-existent data set name or with the REALLOCATE option, as shown in the following example:

```
mvsget 'user1.ftp.ps'  'user1.ftp.exist'   (REALLOCATE
```

For more information about the REALLOCATE option, see the z/OS Communications Server: IP User's Guide and Commands.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

```
proxy mvsget 'user1.ftp.ps'   'user1.ftp.exist'
EZA1701I >>> XDSS 'user1.ftp.ps'
200-LASTREF=2011/11/08 DSEMPTY=FALSE
200 SITE DSNTYPE=BASIC RECFM=VB BLKSIZE=6233 LRECL=256 PRIMARY=1 SECONDARY=1 EAT
TR=SYSTEM
TRACKS

EZA1701I >>> XDSS 'user1.ftp.exist'
200-LASTREF=2011/11/08 DSEMPTY=FALSE
200 SITE PDSTYPE=PDS RECFM=VB BLKSIZE=6233 DIRECTORY=27 LRECL=256 PRIMARY=1 SECO
NDARY=1 EATTR=SYSTEM TRACKS
EZA1779I MVSGET fails: the target data set already exists
EZA1780I To reallocate it, issue the subcommand with the REALLOCATE option
```

EZA1780I To reallocate it, issue the subcommand with the REALLOCATE option

Explanation

This message is issued as part of a message group. See message “EZA1779I” on page 126 for a complete description of the message group.

This message indicates that a data set to be transferred has the same name as a local file. The data set is not transferred.

System action

FTP continues processing.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Rename the local data set and reissue the subcommand, or reissue the subcommand with the REALLOCATE option, as shown in the following example:

```
mvsget 'user1.ftp.ps'  'user1.ftp.exist'   (REALLOCATE
```

For more information about the REALLOCATE option, see the z/OS Communications Server: IP User's Guide and Commands.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCA

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
mvsget 'user1.ftp.ps'   'user1.ftp.exist'
EZA1779I MVSGET fails: the target data set already exists
EZA1780I To reallocate it, issue the subcommand with the REALLOCATE option
EZA1460I Command:
```

EZA1784E **Data to be written exceeds maximum filel LRECL; source ASCII file probably lacks record delimiters.**

Explanation

This message is issued when an FTP GET subcommand converts an ASCII format source file to an EBCDIC format destination file if the number of bytes waiting to be written to the disk exceeds the maximum supported logical record length (LRECL), and the NOWRAPrecord parameter was specified for the LOCSITE command, preventing the client from automatically wrapping longer than allowable records to the next line. The file transfer is not completed. No data is written to disk.

System action

If FTP was invoked with the EXIT option, the session is ended with return code 16150. Otherwise, FTP continues and waits for the next user command.

Operator response

If necessary, restart FTP. Before attempting to transfer the file again, submit the LOCSITE command, specifying the WRAPrecord parameter, allowing the FTP client to wrap longer than allowable records onto a new line, then resubmit the GET command.

System programmer response

No action is needed.

EZA1787W **Not authorized to access requested local file.**

Explanation

The user is not authorized to access the requested local file. Access can be granted by the file owner. Otherwise, the file cannot be accessed.

System action

FTP continues.

Operator response

Contact the administrator for the file and request access.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA1788I ACCOUNT

Explanation

This is a prompt for accounting information from the target host system.

System action

None.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCZ

EZA1865I Command not Supported. Translation Table not Loaded.

Explanation

A valid FTP client TYPE subcommand was entered to change the current transfer type, but the DBCS translation table required for that transfer type is not loaded. The current FTP transfer type is unchanged.

System action

The program continues.

Operator response

Consult the system programmer.

System programmer response

Configure a valid DBCS binary translate table data set in the search order hierarchy for the required DBCS translation table, and specify LOADDBCSTABLES in TCPIP.DATA for the required DBCS translation table. See z/OS Communications Server: IP Configuration Reference for more information about the loading and customizing of DBCS translation tables.

Module

EZAFTPCT

Chapter 4. EZA2*xxxx* messages

EZA2063I **Local directory is partitioned data set *directory***

Explanation

The name of the partitioned data set. A partitioned data set (PDS) is a data set in direct access storage that is divided into partitions called members, each of which can contain a program, part of a program, or data.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2064I **Local directory is *directory***

Explanation

The current local directory.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2065I **Local directory is null string.**

Explanation

No local directory name has been specified.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2070I **Invalid local directory name entered - Too large.**

Explanation

The length of the local directory name is larger than the field specified. The directory name cannot be more than 8 characters in length.

System action

FTP continues.

Operator response

Enter the command with correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2071I **Mismatched quotes on directory name *“name”***

Explanation

You entered a subcommand with *name* as the directory name parameter. The syntax used in the directory name parameter is incorrect.

System action

FTP continues.

Operator response

Enter the subcommand with the correct syntax and try it again. To override the existing directory rather than append a qualifier to that directory, enclose the qualifier in single quotation marks. For example: LCD ‘FTP.FILES’. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2072I **Final directory name is too large.**

Explanation

The directory name is larger than the field specified. The directory name cannot be more than 8 characters in length.

System action

FTP continues.

Operator response

Enter the command with the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

EZA2073I **A qualifier in *pathname* begins with an invalid character.**

Explanation

You entered a subcommand with the pathname parameter *pathname*. The pathname specified in the subcommand is incorrect.

System action

FTP continues.

Operator response

Enter the subcommand with a valid pathname and try it again.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2076I **Directory is already null. ".." operand ignored.**

Explanation

An attempt was made to back up one directory level, when the local directory is already null. The local directory remains the null directory.

System action

FTP continues.

Operator response

None

System programmer response

None

Module

EZAFTPCD

EZA2080I **Data sets will be allocated on *volume***

Explanation

This message displays the list of volumes on which new data sets will be allocated during an FTP session.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2081I **Local directory name set to partitioned data set *directory***

Explanation

The name of the local directory in the partitioned data set. A partitioned data set (PDS) is a data set in direct access storage that is divided into partitions called members, each of which can contain a program, part of a program, or data.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPCL

EZA2082I **Local directory name set to *directory***

Explanation

This message displays the name of the local directory.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2083I **Local directory set to null string.**

Explanation

No local directory has been specified. The local directory field has been set to null. A null string is a character or bit string with a length of 0.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2090I ***directory* is a migrated data set and Noautorecall is specified. Local directory name not changed.**

Explanation

The directory specified is a migrated data set. A migrated data set is a data set that has been moved from one hierarchy of storage to another. The local directory name is not changed. See the z/OS Communications Server: IP User's Guide and Commands for more information about the LCD - Changing the Working Level Qualifier - subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2091I **Error recalling *directory*. Local directory name is not changed.**

Explanation

The directory name specified could not be recalled. The directory name remains unchanged.

System action

FTP continues.

Operator response

Enter the command with the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2092I ***directory* is on a direct access volume that is not mounted and Noautomount is specified.**

Explanation

The local directory remains unchanged because the directory name specified was not found in the mounted direct access volume. The NOAUTOMount parameter can be specified in the SITE or LOCSITE subcommands. NOAUTOMount prevents automatic mounting of direct access volumes.

System action

FTP continues.

Operator response

Enter the command with the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2093I **Error mounting volume for *directory*. Local directory name is not changed.**

Explanation

An unexpected condition occurred in trying to mount the direct access volume for the directory name specified. The local directory name remains unchanged.

System action

FTP continues.

Operator response

Enter the command with the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information about the LCD subcommand.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2097I **No automatic recall of migrated data sets.**

Explanation

No automatic recall has been specified for migrated data sets. The NOAUTORecall parameter can be specified in the SITE and LOCSITE subcommand. NOAUTORecall prevents automatic recall of a migrated hierarchical storage manager (HSM) data set. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2098I **Automatic recall of migrated data sets.**

Explanation

The AUTORecall parameter has been specified. The AUTORecall permits automatic recall of migrated hierarchical storage manager (HSM) data sets. This parameter is specified in the SITE and LOCSITE subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2099I **No automatic mount of direct access volumes.**

Explanation

No automatic mount of direct access volumes has been specified. The NOAUTOMount parameter can be specified in the SITE or LOCSITE subcommands. NOAUTOMount prevents automatic mounting of direct access volumes. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2100I **Automatic mount of direct access volumes.**

Explanation

Automatic mount of direct access volumes has been specified. The AUTOMount parameter can be specified in the SITE or LOCSITE subcommands. AUTOMount permits automatic mounting of direct access volumes. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2101I **Data set mode. (Do not treat each qualifier as a directory.)**

Explanation

The DATASetmode option has been specified for the SITE or LOCSITE subcommand. DATASetmode specifies that all the data set qualifiers below the current directory are treated as entries in the directory. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2102I **Directory mode. (Treat each qualifier as a directory.)**

Explanation

Directory mode has been specified. The directory mode is declared in the Directorymode parameter of the SITE and LOCSITE subcommands. This parameter specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory. In directory mode, this qualifier is only used by the MPUT, MGET, LS, and DIR subcommands. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2103I **FileType is SEQ (Sequential - the default).**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2104I **UTF-8 encoding is used on the control connection**

Explanation

This message indicates UTF-8 encoding is used on the control connection for this session.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

LOCSTAT

EZA2105I **FileType is SQL (Query to perform).**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2106I **srestart *option* not supported for MVS data sets**

Explanation

An **srestart** subcommand was entered to restart a file transfer involving an MVS data set. The **srestart** option is not supported for MVS data sets.

option is the srestart subcommand option entered.

System action

FTP stops processing the current subcommand, and prompts for the next subcommand.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCB

Procedure name

srestart_get(), srestart_put()

EZA2107I **option *option* is not supported for a proxy subcommand**

Explanation

You entered a proxy subcommand that is valid for proxy sessions, but you included a subcommand option that is not supported for proxy sessions.

option is the option that is not supported.

System action

The FTP client rejects the subcommand. Processing continues.

Operator response

Either issue the proxy subcommand without the option, or log in to the proxy server as your primary session and issue the subcommand as a primary session command.

System programmer response

No action is needed.

Module

ezaftpcr

Procedure name

mkdir()

EZA2108I **Confidence=*confidence_level* for *action* of *name***

Explanation

This message is issued to report the level of confidence the FTP client has in the successful completion of a data transfer when CHKCONFIDENCE TRUE is specified or defaulted in the client's FTP.DATA data set.

confidence_level is the level of confidence that the FTP client has in the completion of the transfer. Values for *confidence_level* are:

High
While successful completion of the transfer cannot be guaranteed, the FTP client detected no reason to doubt the successful completion of the transfer.

NoEOF
The FTP client detected a missing EOF marker in a STRU R or MODE B or C inbound file. This level is not reported when another problem is detected.

Low
The FTP client detected a problem with completion of the transfer such as failure of the server to respond after the transfer or another reported error. Low overrides NoEOF if both conditions are present.

action is the action taken on the file or data set. Values for *action* are:

APPEND
File or data set is being appended by the FTP client.

GET
File or data set is being retrieved by the FTP client.

MGET
Files or data sets are being retrieved by the FTP client.

MPUT
Files or data sets are being sent by the FTP client.

PUT
File or data set is being sent by the FTP client.

RESTART (OF GET)
Retrieval of file or data set by FTP client restarted.

RESTART (OF PUT)
Sending of file or data set by FTP client restarted.

SRESTART GET
Stream mode retrieval of file or data set by FTP client restarted.

SRESTART PUT
Stream mode sending of file or data set by FTP client restarted.

name is the name of the file or data set being transferred

System action

FTP continues.

Operator response

Check to see whether the transfer completed successfully and try again if necessary.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

Procedure name

hfs_rcvFile, mvs_rcvFile, sndFile

EZA2109I **Confidence level of data transfers is checked and reported**

Explanation

This message is issued in response to the LOCSTAT subcommand when CHKCONFIDENCE TRUE is coded in the client's FTP.DATA. It means that the confidence level in the successful completion of transfers from or to the FTP client will be determined and reported in message EZA2108I.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2110I **Confidence level of data transfers is neither checked nor reported**

Explanation

This message is issued in response to the LOCSTAT subcommand when CHKCONFIDENCE FALSE is coded in the client's FTP.DATA. This means that the confidence level in successful completion of data transfers will not be determined or reported. This does not disable error reporting.

System action

FTP continues.

Operator response

If you want to activate confidence checking and reporting, see the information about the CHKCONFIDENCE statement in z/OS Communications Server: IP Configuration Reference for information about coding the CHKCONFIDENCE statement in FTP.DATA.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2111I **Outbound SBCS ASCII data uses *lt_chars* line terminator**

Explanation

This message is issued in response to the LOCSTAT subcommand. Each line of single-byte data being transferred to the server and translated to ASCII in the FTP client will use the displayed line terminator. This value is determined by the LOCSITE SBSENDEOL parameter or the SBSENDEOL FTP.DATA statement.

lt_chars is the line terminator that will be used. Values for *lt_chars* are:

CRLF
Single-byte data translated to ASCII will use the combination of a carriage return (x'0D') and line feed (x'0A') to terminate each line. This is the default and the standard line terminator defined by RFC 959. The z/OS server and client can receive ASCII data only in this format.

CR
Single-byte data translated to ASCII will use a carriage return (x'0D') only to terminate each line.

LF
Single-byte data translated to ASCII will use a line feed (x'0A') only to terminate each line.

no
Single-byte data translated to ASCII will use no line terminator characters.

System action

FTP continues.

Operator response

If it is necessary to change the *lt_chars* setting, use the LOCSITE subcommand with the SBSENDEOL parameter, or code an SBSENDEOL statement in FTP.DATA.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2112I **Outbound MBCS ASCII data uses *lt_chars* line terminator**

Explanation

This message is issued in response to the LOCSTAT subcommand. Each line of multi-byte data being transferred to the server and translated to ASCII in the FTP client will use the displayed line terminator. This value is determined by the LOCSITE MBSENDEOL parameter or the MBSENDEOL FTP.DATA statement.

lt_chars is the line terminator that will be used. Values for *lt_chars* are:

CRLF
Multi-byte data translated to ASCII will use the combination of a carriage return (x'0D') and line feed (x'0A') to terminate each line. This is the default and the standard line terminator defined by RFC 959. The z/OS server and client can receive ASCII data only in this format.

CR
Multi-byte data translated to ASCII will use a carriage return (x'0D') only to terminate each line.

LF
Multi-byte data translated to ASCII will use a line feed (x'0A') only to terminate each line.

no
Multi-byte data translated to ASCII will use no line terminator characters.

System action

FTP continues.

Operator response

Change the setting, if necessary, using the LOCSITE MBSENDEOL parameter or the MBSENDEOL FTP.DATA statement.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2117I **Client SBSENDEOL must be CRLF for stream mode restart of put**

Explanation

This message is displayed when the SRESTART PUT subcommand is issued while SBSENDEOL is not set to CRLF. The client cannot reliably restart a PUT operation unless the SBSENDEOL setting was CRLF during the interrupted PUT or APPEND subcommand. Therefore, SRESTART PUT is disabled for the remainder of this FTP login session. See the information about the SBSENDEOL statement in the z/OS Communications Server: IP Configuration Reference for more information about SBSENDEOL settings.

System action

The FTP client stops processing the SRESTART PUT subcommand.

Operator response

The file transfer must be started again from the beginning. If you continue to experience this error, report the error to the system programmer.

System programmer response

Most servers do not require an SBSENDEOL setting other than the default, CRLF. If SBSENDEOL is coded as a different value in FTP.DATA or you use the LOCSITE subcommand to set the value to anything other than CRLF, users will not be able to issue an SRESTART PUT command to restart a file transfer in stream mode. If your users require the alternate SBSENDEOL value for only certain servers, specify SBSENDEOL CRLF in the client FTP.DATA data set and instruct the users to issue the LOCSITE SBSENDEOL subcommand only after logging in to those servers to allow the users to restart outbound stream mode file transfers to the servers that do not require the alternate SBSENDEOL value.

Consider whether block or compressed mode file transfers would be appropriate for your installation. Block and compressed mode transfers can be restarted if you initiated checkpointing before the file transfer. See the information about restarting a failed data transfer in z/OS Communications Server: IP User's Guide and Commands.

Module

EZAFTPCB

Procedure name

None.

EZA2118I **srestart failed - file transferred while SBSENDEOL not CRLF**

Explanation

You issued an SRESTART PUT subcommand to resume a stream mode, type ASCII, SBCS file transfer. Previously, while Mode was Stream, Type was ASCII, Encoding was SBCS, and SBSENDEOL was a value other than CRLF, you sent or tried to send a file with a PUT, MPUT, or APPEND subcommand. The FTP client disabled the SRESTART subcommand for the rest of the session because stream mode file transfer restarts are not reliable under these conditions.

System action

The FTP client prompts for another subcommand.

Operator response

Issue the PUT subcommand to send the file again. If this error occurs often, contact the System Programmer.

System programmer response

For file transfers to servers that do not require an alternate SBSENDEOL value, code the default SBSENDEOL value to be CRLF, and instruct the users to issue the LOCSITE subcommand to change the value only after logging into those servers that support alternate SBSENDEOL values for inbound mode stream, type ASCII file transfers. For file transfers to servers that do require an alternate SBSENDEOL value, do not attempt to SRESTART (stream mode restart) Type ASCII outbound transfers.

See the information about the SBSENDEOL statement in the z/OS Communications Server: IP Configuration Reference for more information about the SBSENDEOL, Mode, Type, and Encoding configuration options.

See the information about the LOCSITE subcommand, the SRESTART subcommand, and the PUT subcommand in z/OS Communications Server: IP User's Guide and Commands.

Module

EZAFTPCB

Procedure name

srestart_put()

EZA2119I **Unrecognized parameter *option* on LOCSTAT subcommand**

Explanation

The LOCSTAT subcommand was entered with a parameter that is not valid.

In the message text:

option
The parameter that was entered.

System action

The subcommand is not processed. FTP continues.

Operator response

Reissue the LOCSTAT subcommand with a valid parameter. See the information about the File Transfer Protocol (FTP) in z/OS Communications Server: IP User's Guide and Commands or transferring files using FTP in z/OS Communications Server: IP Configuration Guide. See the information about the LOCSTat subcommand in z/OS Communications Server: IP User's Guide and Commands for information about the parameters of the LOCSTAT subcommand.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

Not applicable.

Source

Not applicable.

Module

EZAFTPCL

Routing code

Not applicable.

Descriptor code

Not applicable.

Example

Not applicable.

EZA2120I **The value of option *option* is undefined**

Explanation

This message is issued in response to the LOCSTAT subcommand and indicates that the value of the specified option is undefined.

In the message text:

option
The option that was specified on the LOCSTAT subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

Not applicable.

Source

Not applicable.

Module

EZAFTPCL

Routing code

Not applicable.

Descriptor code

Not applicable.

Example

Not applicable.

EZA2121I **Command (*ee-ss-cccc-rrr*):**

Explanation

The FTP client is prompting the FTP callable API for another subcommand. The FTP client is also providing the FTP callable API with information about the previous command processed.

ee is the 2-digit decimal client error code for the subcommand (00 if no error). The FTP Client Error Codes can be found in z/OS Communications Server: IP User's Guide and Commands.

ss is the 2-digit decimal code of the FTP client subcommand (all blanks if none). The FTP subcommand codes can be found in z/OS Communications Server: IP User's Guide and Commands.

cccc is the 4-character FTP command sent to the server (all blanks if no command was sent).

rrr is the 3-digit decimal server reply to the FTP command (all blanks if no command was sent).

The following example is a prompt message that is sent after a GET subcommand completes successfully:

```
EZA2121I Command (00-16-RETR-250):
```

System action

The FTP client waits for the next subcommand.

Operator response

Enter the next subcommand.

System programmer response

No action is needed.

Module

EZAFTPCU

Procedure name

promptCallableAPI

EZA2123I **FtpStop (*ee-ss-cccc-rrr*):**

Explanation

This is the final message sent by the FTP client to the FTP Callable API, which indicates that the FTP client has ended and provides the FTP Callable API with information about the last subcommand (if any).

ee is the 2-digit decimal client error code for the subcommand (00 if no error). The FTP Client Error Codes can be found in z/OS Communications Server: IP User's Guide and Commands.

ss is the 2-digit decimal code of the FTP client subcommand (all blanks if none). The FTP subcommand codes can be found in z/OS Communications Server: IP User's Guide and Commands.

cccc is the 4-character FTP command sent to the server (all blanks if no command was sent).

rrr is the 3-digit decimal server reply to the FTP command (all blanks if no command was sent).

The following example is a message that is sent after a QUIT subcommand completes successfully:

```
EZA2123I FtpStop(00-29-QUIT-221):
```

System action

The FTP client process ends.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCX

Procedure name

stopCallableAPI

EZA2125I **Data sets will be allocated like data set *'name'***

Explanation

All new data sets will be allocated the same as the displayed data set name. The data set name can be specified in the DCbdsn=*data set_name* parameter of the SITE or LOCSITE subcommands. There is no hard-coded default.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2126I **Primary allocation *value* blocks, Secondary allocation *value* blocks.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2127I **Primary allocation *value* blocks, Secondary allocation *value* block.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2128I **Primary allocation *value* block, Secondary allocation *value* blocks.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2129I **Primary allocation *value* block, Secondary allocation *value* block.**

Explanation

The primary and secondary allocations that have been specified for the PRImary=*nnn* and SECondary=*nnn* parameter of the SITE or LOCSITE subcommands. These parameters indicate the amount of primary and secondary allocation. This message also indicates that the BLocks parameter has been declared. This parameter specifies that primary and secondary allocation is in blocks. Depending on the values assigned to these parameters, only one of these messages will be displayed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2130I **Primary allocation *value* cylinders, Secondary allocation *value* cylinders.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2131I **Primary allocation *value* cylinders, Secondary allocation *value* cylinder.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2132I **Primary allocation *value* cylinder, Secondary allocation *value* cylinders.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2133I **Primary allocation *value* cylinder, Secondary allocation *value* cylinder.**

Explanation

The primary and secondary allocations that have been specified for the PRImary=*nnn* and SECondary=*nnn* parameter of the SITE or LOCSITE subcommands. These parameters indicate the amount of primary and secondary allocation. This message also indicates that the CYlinders parameter has been declared. This parameter specifies that primary and secondary allocation is in cylinders. Depending on the values assigned to these parameters, only one of these messages will be displayed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2134I **Primary allocation *value* tracks, Secondary allocation *value* tracks.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2135I **Primary allocation *value* tracks, Secondary allocation *value* track.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2136I **Primary allocation *value* track, Secondary allocation *value* tracks.**

Explanation

This message is displayed when you issue a LOCSTAT command.

EZA2137I **Primary allocation *value* track, Secondary allocation *value* track.**

Explanation

The primary and secondary allocations that have been specified for the PRImary=*nnn* and SECondary=*nnn* parameter of the SITE or LOCSITE subcommands. These parameters indicate the amount of primary and secondary allocation. This message also indicates that the TRacks parameter has been declared. This parameter specifies that primary and secondary allocation is in tracks. Depending on the values assigned to these parameters, only one of these messages will be displayed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2138I **Partitioned data sets will be created with *value* directory blocks**

Explanation

The *value* represents the number of directory blocks to be allocated for the directory of a PDS. This message is issued only if a valid value has been specified.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2139I **Data sets will be allocated using unit *name***

Explanation

All new data sets will be allocated using the displayed data set name as a model.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2140I **FEATURE**

Explanation

This message displays the correct format for the FEATURE subcommand which is used to retrieve status information from a remote host. For more information about the FEATURE subcommand, see the z/OS Communications Server: IP User's Guide and Commands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

usage()

EZA2141I **Number of access method buffers is *value*.**

Explanation

The number of buffers can be specified in the BUFNO=*nn* parameter of the SITE or LOCSITE subcommand. This parameter specifies the number of access method buffers to be used during data transfer. Valid values are in the range of 1 - 255.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2142I **Mgmtclass for new data sets is *value***

Explanation

The value of the management class assigned to newly allocated data sets. The management class can be specified in the MGmtclas=*class* parameter of the SITE or LOCSITE subcommands. This parameter specifies the SMS management class assigned to newly allocated data sets.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2143I **Storclass for new data sets is *value***

Explanation

This message indicates the value of the stor class assigned to newly allocated data sets.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2144I **Dataclass for new data sets is *value***

Explanation

The value of the data class assigned to newly allocated data sets.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2145I **RDW's from VB/VBS files are discarded.**

Explanation

Variable record descriptors (RDW) are discarded from variable-length blocked records and variable-length blocked spanned records. The RDW information can be specified in the NORDW parameter of the SITE or LOCSITE subcommand. This parameter specifies that variable record descriptors (RDWs) are discarded during FTP transmission of variable record length / blocked records (VB) or variable record length / blocked records / spanned records (VBS) data sets in other than block mode.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2146I **RDW's from VB/VBS files are retained as part of data.**

Explanation

The variable record descriptors (RDW) parameter has been declared for the SITE and LOCSITE subcommand. This parameter specifies that RDWs are treated as if they were part of the record and not discarded during FTP transmission of variable record length / blocked records (VB) or variable record length / blocked records / spanned records (VBS) data sets in other than block mode.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2147I **Retention period is *days***

Explanation

The number of days that a newly allocated data set will be retained. The retention period can be specified in the RETPD=*nnn* parameter of the SITE or LOCSITE subcommands. This parameter specifies the number of days that a newly allocated data set should be retained. This value is used to calculate the expiration date associated with the data set.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2148I **DB2® subsystem name is *name***

Explanation

The name of the Db2® subsystem. The Db2 subsystem is specified in the DB2=*db2_name* parameter of the SITE or LOCSITE subcommands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2149I **SQL output sent in SPREADsheet format.**

Explanation

The output generated is sent in spreadsheet format for this SQL file type. This output format is specified in the SPread parameter of the SITE or LOCSITE subcommands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2150I **SQL output sent in report (NOSPREAD) format.**

Explanation

The output generated is not in spreadsheet format for this SQL file type. This information is specified in the NOSPread parameter of the SITE or LOCSITE subcommands. This parameter indicates that the output is not in spreadsheet format when the file type is SQL.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2151I **SQLCOL (column headings) use *name***

Explanation

The SQLCol parameter has been specified in the SITE or LOCSITE subcommands. The SQLCol=*type* parameter indicates the column headings of the output file. The following list describes each type:

Type
Description

Names
Names are the column headings. The labels are ignored. This is the hard-coded default.

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 is the first choice for column heading, but if there is no label, the name becomes the column heading.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2152I **Volid of Migrated Data Sets is *id***

Explanation

The volume ID for allocated new migrated data sets.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2153I **Wrapping data into next record.**

Explanation

Data is being sent as part of the next record.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2154I **Trailing blanks in records read from RECFM F datasets are discarded.**

Explanation

Trailing blanks in records read from fixed record format data sets are discarded.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2155I **Trailing blanks in records read from RECFM F datasets are retained.**

Explanation

Trailing blanks in records read from fixed record format data sets are retained.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2189I **Volume *disk* could not be found.**

Explanation

The volume serial specified in the Qdisk parameter of the LOCSITE subcommand could not be found.

System action

FTP continues.

Operator response

Check for the correct command syntax and volume serial number and try the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2190I **No storage volumes could be found.**

Explanation

The Qdisk parameter was entered without a specific volume serial. When this parameter is left blank, statistics about all storage volumes are displayed. No storage volumes were found.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2192I **Percent Free Free Largest Free**

Explanation

You entered the FTP client LOCSIte subcommand and specified the QDISK parameter to display space statistics for one or more volumes. This message is the first line of a header for a group of messages that is displayed to report space statistics. See message EZA2194I for more information.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

No action is needed.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpmk

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA1460I Command:
locsite qdisk=a72aaa
EZA2192I        Percent      Free      Free          Largest    Free
EZA2193I Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
EZA2194I A72AAA    5          124        23          101    0      5  Private
EZA1460I Command:

locsite qdisk
EZA2192I        Percent      Free      Free          Largest    Free
EZA2193I Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
EZA2194I CPDLB3   58         1937       112         1523    0  99999+ Storage
EZA2194I CPDLB4   77         2570        76         2190    0     16  Storage
EZA2194I CPDLB0   44          486       158          461    0     26  Storage
EZA2194I CPDLB1    1            0       316            0   20     54  Storage
EZA1460I Command:
```

EZA2193I **Volume Free Cyls Trks Cyls-Trks Exts Use Attr**

Explanation

You entered the FTP client LOCSIte subcommand and specified the QDISK parameter to display space statistics for one or more volumes. This message is the second line of a header for a group of messages that is displayed to report space statistics. See message EZA2194I for more information.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

No action is needed.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpmk()

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA1460I Command:
locsite qdisk=a72aaa
EZA2192I        Percent      Free      Free          Largest    Free
EZA2193I Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
EZA2194I A72AAA    5          124        23          101    0      5  Private
EZA1460I Command:

locsite qdisk
EZA2192I        Percent      Free      Free          Largest    Free
EZA2193I Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
EZA2194I CPDLB3   58         1937       112         1523    0  99999+ Storage
EZA2194I CPDLB4   77         2570        76         2190    0     16  Storage
EZA2194I CPDLB0   44          486       158          461    0     26  Storage
EZA2194I CPDLB1    1            0       316            0   20     54  Storage
EZA1460I Command:
```

EZA2194I **EZA2194I *volume_serial percent_free free_cylinders free_tracks l_cylinders l_tracks free_extents use_attribute***

Explanation

You entered the FTP client LOCSIte subcommand and specified the QDISK parameter to display space statistics for one or more volumes. This message is part of a group of messages that is displayed to report space statistics. One EZA2194I message is displayed for each volume that you requested.

In the message text:

volume_serial
The volume serial number of a volume on the FTP client host.

percent_free
The percent of free space on the volume that is identified by the *volume_serial* value.

free_cylinders
The number of free cylinders on the volume that is identified by the *volume_serial* value.

free_tracks
The number of free tracks on the volume that is identified by the *volume_serial* value.

l_cylinders
The number of cylinders in the largest free extent on the volume that is identified by the *volume_serial* value.

l_tracks
The number of tracks in the largest free extent on the volume that is identified by the *volume_serial* value.

free_extents
The number of free extents on the volume that is identified by the *volume_serial* value.

use_attribute
The use attribute of the volume that is identified by the *volume_serial* value.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

No action is needed.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpmk()

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA1460I Command:
locsite qdisk=a72aaa
EZA2192I        Percent      Free      Free          Largest    Free
EZA2193I Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
EZA2194I A72AAA    5          124        23          101    0      5  Private
EZA1460I Command:


locsite qdisk
EZA2192I        Percent      Free      Free          Largest    Free
EZA2193I Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
EZA2194I CPDLB3   58         1937       112         1523    0  99999+ Storage
EZA2194I CPDLB4   77         2570        76         2190    0     16  Storage
EZA2194I CPDLB0   44          486       158          461    0     26  Storage
EZA2194I CPDLB1    1            0       316            0   20     54  Storage
EZA1460I Command:
```

EZA2216W **SQLCOL parameter *(value)* is not valid.**

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.

Operator response

Resubmit the SITE command with a valid value for the SQLCOL parameter. For more information about the SITE command, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2224I **LANGUAGE [language [-subtag]]**

Explanation

This message displays the correct format for the LANGUAGE subcommand. Use the LANGUAGE subcommand to set the language for FTP replies. For more information about the LANGUAGE subcommand, see the z/OS Communications Server: IP User's Guide and Commands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

usage()

EZA2225I **ask the server which features and extensions it supports**

Explanation

This message displays the purpose of the FEATURE subcommand. For more information about the FEATURE subcommand, see the z/OS Communications Server: IP User's Guide and Commands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

usage()

EZA2226I	**request server replies in another language, or reset language to the default**

Explanation

This message displays the purpose of the LANGUAGE subcommand. For more information about the LANGUAGE subcommand, see the z/OS Communications Server: IP User's Guide and Commands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

usage()

EZA2227I	**subcommand rejected. UTF-8 translation table is not available**

Explanation

FTP could not process the current subcommand because the UTF-8 translation table is required to complete the command. The UTF-8 translation table is not available to FTP.

System action

FTP stops processing the current subcommand and prompts the user for the next subcommand.

Operator response

Contact the system programmer with the error.

System programmer response

Verify that the National Language Resources component of z/OS Language Environment® is installed, and that all data sets and files pertinent to UCS-2 and UTF-8 conversion are installed. See the z/OS XL C/C++ Programming Guide.

Module

EZAFTPCR

Procedure name

lang()

EZA2228I **subcommand not available - client is not configured for UTF-8 encoding**

Explanation

FTP cannot process the current subcommand because the subcommand is not valid unless the FTP client is configured for UTF-8 encoding.

System action

FTP stops processing the current subcommand and prompts for the next subcommand.

Operator response

Contact the system programmer with the error message.

System programmer response

Determine whether UTF-8 encoding is appropriate for your installation. If you want UTF-8 encoding, code EXTENSIONS UTF8 in the client's FTP.DATA. See the z/OS Communications Server: IP Configuration Reference for more information about the EXTENSIONS statement.

Module

ezaftpcr

Procedure name

lang()

EZA2229I **UTF-8 encoding has been disabled for this client.**

Explanation

The user issued a LANGuage subcommand. The client is not allowed to send a LANG command to the server unless it supports UTF-8 encoding of the control connection. The user had earlier issued an FTP client LOCSITE CTRLCONN or LOCSITE CCXLATE subcommand, which caused UTF-8 encoding for the control connection to be disabled.

System action

The LANGUAGE subcommand is ignored. FTP continues processing.

Operator response

If you want to use the language subcommand, exit the FTP client program and start the client again, without issuing LOCSITE subcommands to specify a code page for the control connection. If you want the FTP client to use a specific code page for the control connection, you cannot issue the LANGuage subcommand.

System programmer response

No action is needed.

Module

ezaftpcr

Procedure name

language()

EZA2230I **Error allocating storage for catalog search**

Explanation

The command was entered to list MVS files. FTP was unable to get the necessary storage for the catalog list.

System action

Command processing ends. FTP continues with the next command.

Operator response

Try the command again. If the problem persists, contact the system programmer.

System programmer response

Allocate additional storage for the address space.

Module

EZAFTPMR.

Procedure name

PSLIST

EZA2231I **The SOCKS server rejected the connection to *ip_address* with *reply_field***

Explanation

The FTP client attempted to connect to an FTP server. The client is configured to access the FTP server using a SOCKS server. The SOCKS server will not allow the client to connect to the requested FTP server.

ip_address is the IP address of the requested FTP server.

reply_field is a decimal value describing why the SOCKS server rejected the connect request. When SOCKS V4 protocols are in use, the value is the CD field, or result code, from the CONNECT reply as defined in the document *SOCKS: A protocol for TCP proxy across firewalls* by Ying-Da Lee. The following excerpt is from that document listing the possible return codes and explanations.

91
The request rejected or failed.

92
: The request rejected because SOCKS server cannot connect to identd on the client.

93
: The request rejected because the client program and identd report different user IDs.

When SOCKS V5 protocols are in use, the value is the REP field, or reply value, from the CONNECT reply as defined in RFC 1928. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

System action

FTP stops trying to connect to the indicated IP address through the SOCKS server.

Operator response

Contact the system programmer with the error message.

System programmer response

You can obtain more detailed information about why the SOCKS server rejected the connection by using the FTP client debug and dump subcommands to activate tracing, and attempting the connection again.

SOCKS V4 or SOCKS V5 protocols are specified in the file or data set indicated by the SOCKSCONFIGFILE statement in FTP.DATA.

For the SOCKS V5 protocol, the reply value is one of the following:

2
: The SOCKS server is configured to reject attempts to connect to the FTP server. Follow the procedure for SOCKS V4 protocol and CD field 91.

all other reply values
: Contact the administrator of the SOCKS server with the error.

For the SOCKS V4 protocol, the CD field is one of the following:

91
: A probable cause is that the SOCKS server is configured to reject attempts to connect to the FTP server. If you want to allow the FTP client to connect to the target FTP server, either ask the administrator of the configured SOCKS server to allow connections to the FTP server, or change the FTP client SOCKS configuration. You can configure the client to use a different SOCKS server, or you can configure the client to connect to the FTP server without using a SOCKS server.

92, 93
: The SOCKS server requires inetd authentication but inetd authentication is not supported by the FTP client. Change the FTP client SOCKS configuration to use a SOCKS server that does not require identd authentication, or ask the administrator of the SOCKS server to allow the connection without identd authentication.

See the z/OS Communications Server: IP Configuration Guide and the z/OS Communications Server: IP Configuration Reference for information about configuring the FTP client to use SOCKS servers.

Module

ezaftpsc

Procedure name

connectV4(), connectV5().

EZA2232I **SOCKS server rejected authentication method *method***

Explanation

The FTP client tried to connect to an FTP server via a SOCKS server using SOCKS V5 protocols. The SOCKS server rejected the authentication method requested by the client.

method is a hexadecimal value representing the authentication method requested by the FTP client. The values for *method* are defined in RFC 1928. A value of zero signifies NO AUTHENTICATION REQUIRED. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

System action

The FTP client stops trying to connect to the FTP server through the SOCKS server.

Operator response

Contact the system programmer with the error message.

System programmer response

The probable explanation for the rejection is that the SOCKS server does not allow the requested authentication method.

If you want to allow the FTP client to connect to the target FTP server, either ask the administrator of the configured SOCKS server to allow the requested authentication method, or change the FTP client SOCKS configuration. You can configure the client to use a different SOCKS server, or you can configure the client to connect to the FTP server without using a SOCKS server. See the z/OS Communications Server: IP Configuration Guide and the z/OS Communications Server: IP Configuration Reference for information about configuring the FTP client to use SOCKS servers.

Module

ezaftpsc

Procedure name

method()

EZA2233I **The SOCKS server rejected the BIND to *ip_address* with *reply_field***

Explanation

The FTP client tried to establish a data connection to the FTP server through a SOCKS server. The SOCKS server rejected a SOCKS BIND request from the client.

ip_address is the address of the FTP server

reply_field is a decimal value describing why the SOCKS server rejected the BIND request. When SOCKS V4 protocols are in use, the value is the CD field, or result code, from the BIND reply as defined in the document *SOCKS: A protocol for TCP proxy across firewall* by Ying-Da Lee. The following excerpt is from that document listing the possible return codes and explanations.

91
The request rejected or failed.

92
The request rejected because SOCKS server cannot connect to identd on the client.

93
The request rejected because the client program and identd report different user IDs.

When SOCKS V5 protocols are in use, the value is the REP field, or reply value, from the BIND reply as defined in RFC 1928. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

System action

FTP stops trying to access the FTP server through the SOCKS server.

Operator response

Contact the system programmer with the error.

System programmer response

You can get more information about why the SOCKS server rejected the BIND request by using the dump and debug subcommands while attempting the connection. See the z/OS Communications Server: IP User's Guide and Commands for information about the dump and debug subcommands.

SOCKS V4 or SOCKS V5 protocols are specified in the file or data set indicated by the SOCKSCONFIGFILE statement in FTP.DATA.

For the SOCKS V5 protocol, the reply value is one of the following:

2
: The SOCKS server has been configured to reject connections to the target IP address. Follow the procedure for SOCKS V4 protocol and CD field 91.

all other reply values
: Contact the administrator of the SOCKS server with the error.

For the SOCKS V4 protocol, the CD field is one of the following:

91
: A possible explanation is that the SOCKS server is configured to reject connections to the target IP address. If you want to allow the FTP client to connect to the target FTP server, either ask the administrator of the configured SOCKS server to allow connections to the FTP server, or change the FTP client SOCKS configuration. You can configure the client to use a different SOCKS server, or you can configure the client to connect to the FTP server without using a SOCKS server. It is also possible that a general SOCKS server failure occurred. Contact the administrator of the SOCKS server with the error.

92, 93
: The SOCKS server requires inetd authentication but inetd authentication is not supported by the FTP client. Change the FTP client SOCKS configuration to use a SOCKS server that does not require identd authentication, or ask the administrator of the SOCKS server to allow the connection without identd authentication.

See the z/OS Communications Server: IP Configuration Guide and the z/OS Communications Server: IP Configuration Reference for information about configuring the FTP client to use SOCKS servers.

Module

ezaftpsc

Procedure name

bindV4(), bindV5()

EZA2243I **Partitioned data set may not be created within a partitioned data set.**

Explanation

A partitioned data set (PDS) cannot be created within an existing partitioned data set. To create a PDS on the local host, the LMKDIR subcommand is used. A PDS is created with either the fully qualified name or FTP appends the current working directory as the first part of the name.

System action

FTP continues.

Operator response

Enter the command with the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2244I **Data set *name* already exists.**

Explanation

The data set name specified in the *directory* parameter of the MKDIR subcommand already exists.

System action

FTP continues.

Operator response

Check for the correct directory name to be created. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2245I ***data set name* created.**

Explanation

The data set name specified in the LMKDIR subcommand has been created. The LMKDIR creates a partitioned data set (PDS) on the local host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2247I **Volume *number* is not mounted and NoAutoMount specified.**

Explanation

The data set name specified in the LMKDIR subcommand is in a volume that is not mounted. NOAUTOmount has been specified in the SITE or LOCSITE subcommand. The NOAUTOMount parameter prevents automatic mounting of data sets on volumes that are not mounted.

System action

FTP continues.

Operator response

Specify AUTOMount in the SITE and LOCSITE subcommand to permit automatic mounting of data sets on volumes that are not mounted. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2250I ***data set name* is a VSAM data set. DCBDsn parameter ignored.**

Explanation

The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The name specified in DCBDsn is a Virtual Storage Access Method (VSAM) data set.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2251I ***data set name* is not a valid file. DCBDsn parameter ignored.**

Explanation

The data set name specified in the DCBDsn parameter is incorrect. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for the correct data set name and try the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

EZA2252I ***data set name*** **does not exist. DCBDsn parameter ignored.**

Explanation

The data set name specified in the DCBDsn parameter was not found. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for the correct data set name and reissue the command. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2253I **Error recalling *data set name*. DCBDsn parameter ignored.**

Explanation

The File Transfer Protocol (FTP) was unsuccessful in accessing the data set name specified in the DCBDsn parameter. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for the correct data set name and issue the DCBDsn parameter again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2254I ***data set name*** **is migrated data set and Noautorecall is specified.**

Explanation

The data set name specified in the DCBDsn parameter is a migrated data set. A migrated data set is a data set that has been moved from one hierarchy of storage to another. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for the correct data set name and issue the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPGU

EZA2255I **Error mounting volume for** ***data set name*****. DCBDsn parameter ignored.**

Explanation

The File Transfer Protocol (FTP) was unsuccessful in mounting the volume where the data set name specified in the DCBDsn parameter is located. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for the correct volume serial number in the VOLume parameter of the SITE and LOCSITE subcommand, correct the data set name, and issue the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2256I ***data set name*** **is on a direct access volume that is not mounted and Noautomount is specified.**

Explanation

The data set name specified in the DCBDsn parameter is located on a volume that has not been mounted. Noautomount has been specified. The NOAUTOMount parameter prevents automatic mounting of data sets on volumes that are not mounted. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Specify AUTOMount in the SITE and LOCSITE subcommands to permit automatic mounting of data sets on volumes that are not mounted. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2257I ***data set name* is not on a direct access volume. DCBDsn parameter ignored.**

Explanation

The data set name specified in the DCBDsn parameter is not located on a direct access volume. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for the correct volume serial number in the VOLume parameter of the SITE and LOCSITE subcommand, correct the data set name, and issue the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2258I **Error locating file *data set name*. DCBDsn parameter ignored.**

Explanation

The File Transfer Protocol (FTP) was unsuccessful in locating the data set name specified in the DCBDsn parameter. The DCBDsn parameter specifies the name of the data set to be used as a model for allocation of new data sets. The parameter is ignored.

System action

FTP continues.

Operator response

Check for correct command syntax and data set name and issue the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPGU

EZA2269I **Lrecl=X Recfm=VBS BLKSIZE=*size***

Explanation

The record format and block size of a data set specified in the RECfm=*format* and BLocksize=*nnn* parameters of the SITE and LOCSITE subcommands. VBS indicates variable record length and blocked and spanned records. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2270I **Lrecl=X Recfm=VS BLKSIZE=*size***

Explanation

The record format and block size of a data set specified in the RECfm=*format* and BLocksize=*nnn* parameters of the SITE and LOCSITE subcommands. VS indicates variable record length and spanned records. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2279W **BLOCKSIze and LRecl must be equal for *recfm* data sets. BLOCKSIze set to *value*.**

Explanation

The LOCSITE subcommand was entered with one or more of the LRecl, RECFm, and BLOCKSIze parameters, and the resulting parameter values caused a mismatch between the LRecl, RECFm, and BLOCKSIze parameters. The Record Format, *recfm*, requires that LRecl and BLOCKSIze be equal, but the values for BLOCKSIze and LRecl were not the same.

System action

The BLOCKSIze parameter is set to *value*.

Operator response

If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2280W **BLOCKSIze must be a multiple of LRecl for *recfm* data sets. BLOCKSIze set to *value*.**

Explanation

The LOCSITE 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 BLOCKSIze be a multiple of LRecl, but it was not.

System action

The BLOCKSIze parameter is set to *value*.

Operator response

If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2281W **BLOCKSIze must be four larger than LRecl for *recfm* data sets. LRecl set to *value*.**

Explanation

The LOCSITE subcommand was entered with one or more of the LRecl, RECFm, and BLOCKSIze parameters, and the resulting parameter values caused a mismatch between the LRecl, RECFm, and BLOCKSIze parameters. The Record Format, *recfm*, requires that BLOCKSIze be at least 4 more than LRecl, but the value specified for LRecl would not allow BLOCKSIze to be four more than LRecl and still be within its valid range of values.

System action

The LRecl parameter is set to *value*.

Operator response

If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2282W **BLOCKSIze must be four larger than LRecl for *recfm* data sets. LRecl set to *value*. BLOCKSIze set to *value*.**

Explanation

The LOCSITE subcommand was entered with one or more of the LRecl, RECFm, and BLOCKSIze parameters, and the resulting parameter values caused a mismatch between the LRecl, RECFm, and BLOCKSIze parameters. The Record Format, *recfm*, requires that BLOCKSIze be at least four more than LRecl, but the value specified for LRecl would not allow BLOCKSIze to be four more than LRecl and still be within the valid range of values.

System action

The BLOCKSIze parameter is set to *value*.

Operator response

If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2283W **BLOCKSIze must be four larger than LRecl for *recfm* data sets. LRecl set to *value*. BLOCKSIze set to *value*.**

Explanation

The LOCSITE subcommand was entered with one or more of the LRecl, RECFm, and BLOCKSIze parameters, and the resulting parameter values caused a mismatch between the LRecl, RECFm, and BLOCKSIze parameters. The Record Format, *recfm*, requires that BLOCKSIze be at least 4 more than LRECL, but the value of BLOCKSIze did not meet this criteria.

System action

The BLOCKSIze parameter is set to *value*.

Operator response

If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2284I ***value***

Explanation

The value of the output buffer. This buffer contains characters in ASCII form that are being read from the File Transfer Protocol (FTP) connection.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2294I **Proxy connected through SOCKS server at *IP_address***

Explanation

The user issued a LOCSTat FTP subcommand to display client status. The client has a proxy session established with an FTP server through a SOCKS server.

For a connection through a SOCKS server, the FTP client always connects to port 1080 of the SOCKS server's *IP_address* instead of connecting to the IP address and port number of the FTP server; the SOCKS server relays data between the FTP client and the FTP server. The IP address and port number of the FTP proxy server are also displayed by the LOCSTat subcommand.

IP_address is the IP address of the SOCKS server.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

ezaftpcl

Procedure name

locstat()

EZA2295I **Partitioned data sets will be created in extended format**

Explanation

This message is issued when a valid value has been specified and the PDSTYPE is PDSE. The PDSTYPE statement in FTP.DATA, and the PDSTYPE parameter of the LOCSIte subcommand, are described in z/OS Communications Server: IP User's Guide and Commands.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

Procedure name

locstat()

EZA2300I **Command ended because the CPU time limit was exceeded**

Explanation

The FTP client was processing a command when the process was stopped either by the z/OS UNIX shell command kill -s SIGXCPU or because the CPU time limit (MAXCPUTIME in BPXPRM*nn* or CPUTIMEMAX in the user OMVS segment) was exceeded.

System action

The FTP client ends.

Operator response

Contact the system programmer.

System programmer response

If the process was stopped because the CPU time limit was exceeded, check the MAXCPUTIME statement in the BPXPRM*nn* member or the CPUTIMEMAX statement in the OMVS segment of the user. See the information about the MAXCPUTIME statement in z/OS UNIX System Services Planning and BPXPRMxx member in z/OS MVS Initialization and Tuning Reference.

Module

EZAFTPCX

Procedure name

pgmxcpuc

EZA2301E **Cannot open FTP.DATA specified by the -f parameter *filename***

Explanation

The FTP client was started with the -f filename parameter on the FTP command. FTP tried to open that file or data set but could not.

In the message text:

filename
The name of the FTP.DATA file or data set.

System action

The FTP client exits.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Specify a valid file or data set name and start the FTP client.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPCY

Routing code

1

Descriptor code

2

Example

```
EZA2301E Cannot open FTP.DATA specified by -f parameter /u/user79/not.there
```

EZA2320I ***temp*** **is a migrated data set and Noautorecall is specified.**

Explanation

The named file was sent with AUTORECALL set to false. This prevents automatic recall of the migrated Hierarchical Storage Manager (HSM) data sets.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2321I **Error recalling** ***temp*** **(rc=*****value*****)**

Explanation

The file displayed cannot be recalled.

System action

FTP rejects the current subcommand. Processing continues.

Operator response

Reissue the command. If the problem persists, contact the system programmer.

System programmer response

Additional messages might be issued by your hierarchical storage manager (HSM) product. See the documentation for that product to interpret those messages.

Module

EZAFTPCG, EZAFTPCP

EZA2322I ***temp* is on a direct access volume that is not mounted and Noautomount is specified.**

Explanation

The value for the FTP AUTOMOUNT parameter is set to NOAUTOMOUNT, preventing automatic mounting of volumes. The data set *temp* is on a volume that has not been mounted.

System action

FTP rejects the current subcommand. Processing continues.

Operator response

If possible, issue the **LOCSIte** subcommand to change the automount setting. Otherwise, ask the system programmer to mount the volume containing *name*. Issue the subcommand again. See z/OS Communications Server: IP User's Guide and Commands for information about the **LOCSIte** subcommand.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2323I **Error mounting volume for *temp***

Explanation

FTP is unable to mount the volume for the data set *temp*.

System action

FTP rejects the current subcommand. Processing continues.

Operator response

Issue the subcommand again. If the problem persists, contact the system programmer with the error.

System programmer response

Determine why the requested volume cannot be mounted to the MVS system.

Module

EZAFTPCD, EZAFTPCG, EZAFTPCP

EZA2326I **Error allocating data set *name***

Explanation

The data set name displayed is not valid.

System action

FTP continues.

Operator response

Check the DCBDSN statement in the *hlq*.FTP.DATA data set and verify that the data set name is correct. For more information on the DSBDSN, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

FTP continues.

Module

EZAFTPCD

EZA2329I **No files match the pattern *name***

Explanation

There are no files or data sets that match the pattern.

System action

FTP continues.

Operator response

Try the command again with the correct pattern.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2331I **“*name*” contains an asterisk (“*”) but it is not the last character.**

Explanation

The data set name is not valid.

System action

FTP continues.

Operator response

Check the z/OS Communications Server: IP User's Guide and Commands for information on naming requirements of data sets. Try the command again with the correct name.

System programmer response

No action is needed.

EZA2332I **Patterns (“%” and “*”) may only be specified in the last qualifier when in directory mode.**

Explanation

DIRECTORYMODE specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory. Since you are not in DIRECTORYMODE, pattern matching cannot be used.

System action

FTP continues.

Operator response

If you would like to use pattern matching, switch to DIRECTORYMODE. Check the z/OS Communications Server: IP User's Guide and Commands for information on setting FTP parameters.

System programmer response

No action is needed.

EZA2333I **** must be preceded and followed by a period or blank**

Explanation

The LIST (DIR) or NLST (LS) command was entered with the ** global file-name character in the pathname, but the ** was either not preceded by or not followed by a period (.) or blank. The ** character is used to replace zero or more complete qualifiers in the name.

System action

FTP continues.

Operator response

See the z/OS Communications Server: IP User's Guide and Commands for the rules for using special characters in LIST and NLST commands.

System programmer response

No action is needed.

EZA2337I ***temp* is not a valid file.**

Explanation

The file displayed is not valid.

System action

FTP continues.

Operator response

Reissue the command with a valid file name.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2338I ***temp* is not a partitioned data set.**

Explanation

The name displayed is not a partitioned data set.

System action

FTP continues.

Operator response

Reissue the command with the correct data set name. For more information on partitioned and sequential data sets, check the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2339I ***temp* is a non–existent partitioned data set. Use LMKDIR to create it.**

Explanation

The data set name displayed does not exist. Create the data set using LMKDIR.

System action

FTP continues.

Operator response

To create a partitioned data set, type LMKDIR followed by the data set name. Naming requirements for a PDS are as follows:

- No part of the name can start with a number.
- No part of the name can be more than 8 characters in length.
- Each part of the name is separated by a period.
- If single quotation marks are not used when specifying the PDS name, FTP appends the current working directory as the first part of the name.

For more information on naming conventions for partitioned data sets, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

FTP continues.

Module

EZAFTPCG

EZA2340I ***temp* is a VSAM data set. VSAM is not supported.**

Explanation

The data set name displayed is a Virtual Storage Access Method (VSAM) data set. VSAM data sets are not supported by TCP/IP.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2358W **Unrecognized option '*parameter=operand*' on locsite command.**

Explanation

The LOCSITE subcommand was entered with the option *parameter*, but *parameter* was not a valid LOCSITE subcommand parameter.

System action

The parameter is ignored. FTP continues.

Operator response

Reissue the LOCSITE subcommand with a valid parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2359W **Operand required on *parm* option. Option ignored.**

Explanation

The option *parm* appeared on a LOCSITE subcommand without an operand. The option is ignored.

System action

FTP continues.

Operator response

If wanted, reissue the LOCSITE command with an operand for the *parm* option. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guidefor information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2360W **Operand not allowed on *parm* option. Option ignored.**

Explanation

The parameter *parm* appeared on a LOCSITE subcommand in incorrect format. No operand is allowed for this parameter. The parameter is ignored.

System action

FTP continues.

Operator response

If wanted, reissue the LOCSITE command without an operand on the *parm* parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guidefor information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2370W **DCbdsn already specified. "*parameter=value*" ignored.**

Explanation

DCbdsn parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2372W **LRecl parameter already specified. "*parameter=value*" ignored.**

Explanation

LRecl parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2373W **LRecl parameter *(value)* is not numeric.**

Explanation

The LOCSITE command was specified with the LRecl parameter, but the value, *value*, specified for the LRecl parameter was not a numeric value.

System action

The LRecl parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2374W **PRImary parameter already specified. “*parameter=value*” ignored.**

Explanation

PRImary parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2375W **PRImary parameter *(value)* is not numeric.**

Explanation

The LOCSITE 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.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2376W **SECondary parameter already specified. '*parameter=value*' ignored.**

Explanation

SECondary parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2377W **SECondary parameter *(value)* is not numeric.**

Explanation

The LOCSITE 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.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2378W **VOLume parameter already specified. "*parameter=value*" ignored.**

Explanation

VOLume parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2379W ***value* is not a valid VOLume parameter.**

Explanation

The LOCSITE command was specified with the VOLume parameter, but the value, *value*, specified for the VOLume parameter was not a valid value. The parameter may not be more than 6 characters.

System action

The VOLume parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2380W **RECFm parameter already specified. "*parameter=value*" ignored.**

Explanation

RECFm parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2381W **RECFm parameter *(value)* is not valid.**

Explanation

The LOCSITE command was specified with the RECFm parameter, but *value*, specified for the RECFm parameter, was not a valid value.

System action

The RECFm parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2383W **BLOCKSIze parameter already specified. "*parameter=value*" ignored.**

Explanation

BLOCKSIze parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2384W **BLOCKSIze parameter *(value)* is not numeric.**

Explanation

The LOCSITE command was specified with the BLOCKSIze parameter, but the value, *value*, specified for the BLOCKSIze parameter was not a numeric value.

System action

The SECondary parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2385W **DIRectory parameter already specified. "*parameter=value*" ignored.**

Explanation

DIRectory parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2386W **Directory parameter *(value)* is not numeric.**

Explanation

The LOCSITE 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.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2388W ***value* is not a valid Qdisk parameter.**

Explanation

The LOCSITE command was specified with the Qdisk parameter, but the value, *value*, specified for the Qdisk parameter was not a valid value. The parameter may not be more than 6 characters.

System action

The Qdisk parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2394I **Invalid format data set name “*name*”. DCBDSN parameter ignored.**

Explanation

The data set name displayed is not valid, and therefore is ignored.

System action

FTP continues.

Operator response

The DCBDSN parameter specifies the name of the data set to be used as a model for allocation of new data sets. Reissue the command with the correct data set name. For more information on DCBDSN, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCK, EZAFTPGU

EZA2395W **LRecl must be between 0 and 32760. LRecl parameter ignored.**

Explanation

The record size should be between 0 and 32760. The default is 256.

System action

FTP continues.

Operator response

Reissue the LOCSITE LRecl command with a record size between 0 and 32760.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2396W **BLOCKSIze must be between 0 and 32760. BLOCKSIze parameter ignored**

Explanation

The block size of a data set should be between 0 and 32760. The default is 6233.

System action

FTP continues.

Operator response

Reissue the LOCSITE BLOCKSIze command with a block size between 0 and 32760.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2397W **Unit parameter already specified. *parameter* ignored.**

Explanation

The unit name specifying the unit type for allocation of data sets has already been set. The unit name displayed is ignored.

System action

FTP continues.

Operator response

To change the unit type, specify the unit name in the *hlq*.FTP.DATA data set. The unit name parameter can be used to specify either direct access or tape device types.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2398W ***parameter* is not a valid unit parameter.**

Explanation

The unit parameter is not valid.

System action

FTP continues.

Operator response

Reissue the unit name command with the correct parameter.

System programmer response

FTP continues.

Module

EZAFTPCK

EZA2399W **FILEtype parameter already specified. “*parameter*=*value*” ignored.**

Explanation

The LOCSITE subcommand was entered with the filetype parameter more than once. The file type displayed is ignored.

System action

FTP continues.

Operator response

Issue another LOCSIte subcommand to change the data set file type. Specify the filetype value you want to use on the filetype parameter. see the z/OS Communications Server: IP User's Guide and Commands for more information about the LOCSIte subcommand and the filetype parameter. The default file type is SEQ.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2400W **FileType parameter *parameter* is not valid.**

Explanation

The file type displayed is not a valid file type for a data set.

System action

FTP continues.

Operator response

Reissue the FILETYPE command with a valid file type. The default is SEQ (sequential data set). See message EZA2399 for a list of valid file types.

System programmer response

No action is needed.

Module

EZAFTPCK, EZAFTPEP

EZA2403I **Using control connection address *ipaddr* for data connection**

Explanation

The client sent a PASV command to the server to establish a data connection. By default, an FTP client connects to the IP address and port number specified in the PASV reply to complete the data connection. However, because PASSIVEIGNOREADDR is set to TRUE, the FTP client ignores the IP address in the PASV reply. The FTP client will connect to the port number specified in the PASV reply, at the IP address it used to log into the server, to complete the data connection.

In the message text:

ipaddr
The IP address that the FTP client used when it logged in to the FTP server.

System action

The FTP client ignores the IP address in the PASV reply from the server and connects to the same address that it used to log in to the server.

Operator response

Not applicable.

System programmer response

Not applicable.

User response

No response is necessary if you intend for the FTP client to use the same IP address for the data connection that it used to log in to the FTP Server. If you want the client to use the IP address specified in the PASV reply, configure PASSIVEIGNOREADDR FALSE by modifying the client FTP.DATA file, or by issuing the LOCSIte subcommand with the NOPASSIVEIGNOREADDR parameter.

See the information about the PASSIVEIGNOREADDR (FTP client) statement in z/OS Communications Server: IP Configuration Reference. See the information about the LOCSIte subcommand in z/OS Communications Server: IP User's Guide and Commands.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

EZAFTPSC

Routing code

*

Descriptor code

*

Automation

Not applicable for automation.

Example

```
Command:
put /etc/hosts
>>> PASV
227 Entering Passive Mode (129,42,17,103,19,136)
EZA2403I Using control connection address 9.155.11.97 for data connection
>>> STOR hosts
125 Storing data in the Null directory (*dev.null).
250 Transfer completed successfully.
41 bytes transferred in 0.005 seconds.  Transfer rate 8.20 Kbytes/sec.
Command:
```

EZA2406W **MGmtclass parameter already specified. “*parameter*=*value*” ignored.**

Explanation

The MGmtclass parameter has already been specified. The parameter displayed is ignored.

System action

FTP continues.

Operator response

The Storage Management System (SMS) has been used to set certain defaults for the *hlq*.FTP.DATA data set. The defaults for space and RECFM are set by MGmtclass. The values for these parameters take precedence over any values you specify using the individual storage attributes parameter with the LOCSITE command or the *hlq*.FTP.DATA data class. To change any of the parameters, see your systems administrator.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2407W **Mgmtclass parameter *parameter* is not valid.**

Explanation

The MGmtclass parameter displayed is not valid.

System action

FTP continues.

Operator response

The Storage Management System (SMS) has been used to set certain defaults for the *hlq*.FTP.DATA data set. The defaults for space and RECFM are set by MGmtclass. The values for these parameters take precedence over any values you specify using the individual storage attributes parameter with the LOCSITE command or the *hlq*.FTP.DATA data class. To change any of the parameters, see your systems administrator.

System programmer response

No action is needed.

EZA2408W **STOrclass parameter already specified. "*parameter*=*value*" ignored.**

Explanation

The STOrclass parameter has already been specified. The parameter displayed is ignored.

System action

FTP continues.

Operator response

The Storage Management System (SMS) has been used to set certain defaults for the *hlq*.FTP.DATA data set. The defaults for unit and DCBDSN are set by STOrclass. The values for these parameters take precedence over any values you specify using the individual storage attributes parameter with the LOCSITE command or the *hlq*.FTP.DATA data class. To change any of the parameters, see your systems administrator.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2409W **STORclass parameter (*parameter*) is not valid.**

Explanation

The STORclass parameter displayed is not valid.

System action

FTP continues.

Operator response

The Storage Management System (SMS) has been used to set certain defaults for the *hlq*.FTP.DATA data set. The defaults for unit and DCBDSN are set by STORclass. The values for these parameters take precedence over any values you specify using the individual storage attributes parameter with the LOCSITE command or the *hlq*.FTP.DATA data class. To change any of the parameters, see your systems administrator.

System programmer response

No action is needed.

EZA2410W **DATAClass parameter already specified. “*parameter*=*value*” ignored.**

Explanation

The DATAClass parameter has already been specified. The parameter displayed is ignored.

System action

FTP continues.

Operator response

The Storage Management System (SMS) has been used to set certain defaults for the *hlq*.FTP.DATA data set. The defaults for space, RECFM, BLKSIZE, and LRECL are set by DATACLASS. The values for these parameters take precedence over any values you specify using the individual storage attributes parameter with the LOCSITE command or the *hlq*.FTP.DATA data class. To change any of the parameters, see your systems administrator.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2411W **DATAClass parameter *parameter* is not valid.**

Explanation

The DATAClass parameter displayed is not valid.

System action

FTP continues.

Operator response

The Storage Management System (SMS) has been used to set certain defaults for the *hlq*.FTP.DATA data set. The defaults for space, RECFM, BLKSIZE, and LRECL are set by DATAClass. The values for these parameters take precedence over any values you specify using the individual storage attributes parameter with the LOCSITE command or the *hlq*.FTP.DATA data class. To change any of the parameters, see your systems administrator.

System programmer response

No action is needed.

EZA2412W **NoRDW already specified. RDW ignored.**

Explanation

RDW (record descriptor word) has been set to false and therefore, discarded during FTP transmission of VB (variable blocked) or VBS (variable blocked spanned) data sets in other than block mode.

System action

FTP continues.

Operator response

Set RDW to true in the *hlq*.FTP.DATA data set if you do not want the RDW to be discarded.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2413W **RDW already specified. NoRdw ignored.**

Explanation

Setting RDW to true specifies that the record descriptor words are treated as if they were part of the record and not discarded during FTP transmission of VB (variable block) or VBS (variable blocked spanned) data sets in other than block mode. RDW information is stored in binary. Transfer files in binary mode to avoid translation problems that can occur if you transfer this binary field in EBCDIC or ASCII.

System action

FTP continues.

Operator response

To discard RDW during FTP transmission, set RDW to false in the *hlq*.FTP.DATA data set.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2414W **RETpd parameter already specified. “*parameter=value*” ignored.**

Explanation

The RETpd parameter has already been set, therefore, the parameter displayed is ignored.

System action

FTP continues.

Operator response

The RETpd parameter for MVS specifies the number of days that a newly allocated data set should be retained. The maximum is 9999.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2415W **RETpd parameter *(value)* is not valid.**

Explanation

The RETpd parameter displayed is not valid.

System action

FTP continues.

Operator response

The RETpd parameter for MVS specifies the number of days that a newly allocated data set should be retained. The maximum is 9999. Reissue the parameter with a valid numeric.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2416W **DB2 parameter *(db2name)* is more than 4 characters.**

Explanation

The *db2name* specified for the DB2 parameter on an FTP LOCSITE command is too long.

System action

The DB2 parameter is ignored. FTP continues.

Operator response

Correct the name of the Db2 subsystem, and issued the LOCSITE command again.

System programmer response

None

Module

EZAFTPCK

EZA2417W **READTAPEFormat *value* is not supported - READTAPEFormat ignored**

Explanation

The *value* specified for the READTAPEFormat parameter on a LOCSITE command is not supported. Valid values are F (for Fixed), V (for Variable), S (for Spanned), X (for Lrecl X), or blank (unspecified).

System action

The READTAPEFormat parameter is ignored. FTP continues.

Operator response

Change value to one of the supported formats.

System programmer response

No action is needed.

Module

EZAFTPCK

Procedure name

locsite

EZA2421W **NORESTGet already specified. RESTGet ignored.**

Explanation

NORESTGet cmd was followed by a RESTGet cmd within the same LOCSITE request.

System action

FTP continues.

Operator response

If the setting is not what you want, reissue the LOCSITE command with the correct values.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2422W **RESTGet already specified. NORESTGet ignored.**

Explanation

RESTGet cmd was followed by a NORESTGet cmd within the same LOCSITE request.

System action

FTP continues.

Operator response

If the setting is not what you want, reissue the LOCSITE command with the correct values.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2423W **CHKPTPREFIX *value* must be Home, Local or Userid.**

Explanation

While processing the FTP.DATA file, the server encountered an incorrect CHKPTPREFIX parameter. The valid values for the CHKPTPREFIX parameter are:

- Home
- Local
- Userid

System action

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

Operator response

Contact the System programmer with the error message to have the FTP.DATA file corrected.

System programmer response

Correct the CHKPTPREFIX parameter in the FTP.DATA file.

Module

EZAFTPSM

Procedure name

read_ftpdata()

EZA2424W **CHKptint parameter (*parameter*) is not valid**

Explanation

The value specified for the CHKptint parameter of the LOCSITE command contained non-numeric characters.

System action

FTP continues.

Operator response

Reissue the LOCSITE command with valid values.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2428I	**CHKPTPrefix uses *value* to determine the HLQ of the FTP.CHECKPOINT file**

Explanation

This is the result of a LOCSTAT command.

One of three possible valid values for CHKPTPREFIX:

Home
The TSO prefix or the z/OS UNIX file system path

Userid
The FTP user ID

Local
The current directory from the LCD command

System action

None.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2477I	**Secure port *port_number* does not allow the *subcommand_name* subcommand**

Explanation

The user entered an FTP subcommand while the FTP client was connected to the secure port. The secure port is the port number defined by the TLSPORT statement in the client's FTP.DATA file. FTP connections to the secure port are implicitly secured. The subcommand is not valid for TLS-secured sessions that are implicitly secured by connecting to the server using the TLSPORT port.

In the message text:

port_number
The port number configured with the TLSPORT statement in the client's FTP.DATA file, or the default TLSPORT number when no TLSPORT statement was coded in FTP.DATA.

subcommand_name
The subcommand that the user entered.

System action

The subcommand is ignored.

Operator response

No action is needed.

System programmer response

If the subcommand is necessary on a TLS-secured session, do not connect to the TLSPORT. Instead, perform the steps for customizing the FTP client for TLS in z/OS Communications Server: IP Configuration Guide to secure the session for TLS without using the TLSPORT. See the information about the TLSPORT FTP client and server in z/OS Communications Server: IP Configuration Reference for more information about the TLSPORT statement. See the z/OS Communications Server: IP User's Guide and Commands for more information about subcommands.

User response

Contact the system programmer.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: FTP

Module

Not applicable.

Routing code

10

Descriptor code

12

Example

In this example, assume that no TLSPORT statement is coded in the client's FTP.DATA file, but that the client has connected to the FTP server at port 990, the default value for the TLSPORT statement.

The user enters the **ccc** subcommand in response to the FTP client prompt:

```
Command: ccc
EZA2477I Secure port 990 does not allow the ccc subcommand.
```

EZA2491I **Compressed mode implemented with type EBCDIC only.**

Explanation

Compressed file transfers are implemented in EBCDIC format only.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2494I **Checkpoint interval is *interval***

Explanation

The interval for generating checkpoint markers is displayed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2500W **No valid checkpoint data available**

Explanation

A RESTART command was issued, but there is no valid checkpoint data available.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCB

EZA2501W **Restart requires EBCDIC data type**

Explanation

A RESTART command was issued, but the representation type is not EBCDIC.

System action

FTP continues.

Operator response

Change the representation type to EBCDIC and reissue the RESTART command.

System programmer response

No action is needed.

Module

EZAFTPCB

EZA2502W **Restart requires Block or Compressed transfer mode**

Explanation

A RESTART command was issued, but the transfer mode is not block or compressed.

System action

FTP continues.

Operator response

Change the transfer mode to block or compressed and reissue the RESTART command.

System programmer response

No action is needed.

Module

EZAFTPCB

EZA2508E **Block or Compressed Mode requires Type E (EBCDIC)**

Explanation

Block or compressed mode file transfer requires a transfer type of EBCDIC. No other transfer type is supported for block or compressed mode file transfer.

System action

No files are transferred.

Operator response

Use the TYPE command to change the transfer type to E (EBCDIC), or the MODE command to change the mode to stream. For more information about these commands, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2509I ***number*** **megabytes transferred -** ***interval*** **second interval rate** ***num_transferred_interval*** **KB/sec - Overall transfer rate** ***num_transferred_overall*** **KB/sec**

Explanation

This message indicates the total number of bytes of binary data transferred to or from the FTP client and the current rate of transfer.

number is the total number of megabytes transferred since the transfer started. Megabyte is defined as 1024*1024 bytes.

interval is the length of the measured transfer interval in seconds. This rate is set using LOCSITE PROGRESS or the PROGRESS statement in the FTP.DATA file.

num_transferred_interval is the number of KBs (KB equals 1000 bytes) transferred per second in this interval.

num_transferred_overall is the number of KBs (KB equals 1000 bytes) transferred per second since the transfer started.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2511I **Checkpoint data set will be opened for GET**

Explanation

The message is issued in response to a LOCSTAT subcommand to indicate that a checkpoint data set will be opened for all GET subcommands if type is E and mode is B or C.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2512I **Checkpoint data set will not be opened for GET**

Explanation

The message is issued in response to a LOCSTAT subcommand to indicate that a checkpoint data set will not be opened for any GET subcommands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2515W **NOTRUNcate already specified. TRUNcate ignored.**

Explanation

A LOCSITE command was issued with both NOTRUNCATE and TRUNCATE options specified. Only the first option is accepted.

System action

None.

Operator response

Reenter the LOCSITE command to change the option.

System programmer response

No action is needed.

Module

EZAFTPCK

Procedure name

locsite()

EZA2516I **Truncated records will be treated as an error and**

Explanation

This message is issued as the result of the LOCSTAT command when NOTRUNCATE is set. If a record is discovered that is longer than LRECL and NOWRAPRECORD is set, an error will be returned and the file transfer will fail.

System action

None.

Operator response

If this is not the intended result, issue the LOCSITE TRUNcate command.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2517I **the file transfer will fail.**

Explanation

This message is issued as the result of the LOCSTAT command when NOTRUNCATE is set. If a record is discovered that is longer than LRECL and NOWRAPRECORD is set, an error will be returned and the file transfer will fail.

System action

None.

Operator response

If this is not the intended result, issue the locsite TRUNcate command.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2518I **Records on input tape are unspecified format**

Explanation

This message is part of the LOCSTAT output. It means that no value has been specified for READTAPEFormat and the record format for an input tape will not be known until the data set is opened.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2519I **Records on input tape are fixed format**

Explanation

This message is part of the LOCSTAT output. It means that a value of F was specified for READTAPEFormat and the record format for an input tape is expected to be fixed. This value must be consistent with the tape label when the data set is opened.

System action

FTP continues

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2520I **Records on input tape are variable format**

Explanation

This message is part of the LOCSTAT output. It means that a value of V was specified for READTAPEFormat and the record format for an input tape is expected to be variable. This value must be consistent with the tape label when the data set is opened.

System action

FTP continues

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2521I **Records on input tape are spanned format**

Explanation

This message is part of the LOCSTAT output. It means that a value of S was specified for READTAPEFormat and the record format for an input tape is expected to be spanned. This value must be consistent with the tape label when the data set is opened.

System action

FTP continues

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2522I **Records on input tape are lrecl X format**

Explanation

This message is part of the LOCSTAT output. It means that a value of X was specified for READTAPEFormat and the record format for an input tape is expected to be lrecl X. This value must be consistent with the tape label when the data set is opened.

System action

FTP continues

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2523E **READTAPEFormat value is *value* but input tape is *format***

Explanation

A put was issued for a data set on tape and a non-blank value was specified for READTAPEFormat. The READTAPEFormat specification does not match the tape label.

value can be one of the following:

F
: Format was to be fixed

V
: Format was to be variable

S
: Format was to be spanned

X
: Format was to be lrecl X

If *value* is X, *format* is:

lrecl *nnnnn* — tape label shows logical record length *nnnnn*

If *value* is F, V, or S, *format* is:

lrecl X — tape label shows lrecl X format

System action

The request is rejected. FTP continues.

Operator response

Either specify the matching READTAPEFormat or change it to unspecified and try the request again. See the z/OS Communications Server: IP User's Guide and Commands for the description of the LOCSITE command for guidelines on using READTAPEFormat.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndFile

EZA2524E **READTAPEFormat value is *value* but input tape is spanned**

Explanation

A put was issued for a data set on tape and a non-blank value was specified for READTAPEFormat. The READTAPEFormat specification does not match the tape label.

value is one of the following:

F
: Format was to be fixed

V
: Format was to be variable

System action

The request is rejected. FTP continues.

Operator response

Either specify the matching READTAPEFormat or change it to unspecified and try the request again. See the z/OS Communications Server: IP User's Guide and Commands for the description of the LOCSITE command for guidelines on using READTAPEFormat.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndFile

EZA2525E **READTAPEFormat value is *value* but input tape is not spanned**

Explanation

A put was issued for a data set on tape and a non-blank value was specified for READTAPEFormat. The READTAPEFormat specification does not match the tape label.

value is one of the following:

S
: Format was to be spanned

System action

The request is rejected. FTP continues.

Operator response

Either specify the matching READTAPEFormat or change it to unspecified and try the request again. See the z/OS Communications Server: IP User's Guide and Commands for the description of the LOCSITE command for guidelines on using READTAPEFormat.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndFile

EZA2526E **READTAPEFormat value is *value* but input tape is format *format***

Explanation

A put was issued for a data set on tape and a non-blank value was specified for READTAPEFormat. The READTAPEFormat specification does not match the tape label.

value can be one of the following:

F
: Format was to be fixed

V
: Format was to be variable

format is the record format string determined from the tape label, such as VB, FBA, etc.

System action

The request is rejected. FTP continues.

Operator response

Either specify the matching READTAPEFormat or change it to unspecified and try the request again. See the z/OS Communications Server: IP User's Guide and Commands for the description of the LOCSITE command for guidelines on using READTAPEFormat.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndFile

EZA2527W **TRUNcate already specified. NOTRUNcate ignored.**

Explanation

Both the TRUNCATE and NOTRUNCATE options have been entered on a LOCSITE command. Only the first option is accepted.

System action

None.

Operator response

Reenter the LOCSITE command to change the option.

System programmer response

No action is needed.

Module

EZAFTPCK

Procedure name

locsite()

EZA2529I **Truncated records will not be treated as an error.**

Explanation

This message will be issued as the result of the LOCSTAT command when the TRUNCATE option is set. The WRAPRECORD option controls whether records are truncated or wrapped into the next record. The file transfer continues.

System action

None.

Operator response

If this is not the intended result, issue the LOCSITE NOTRUNcate command.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2530I **Record format: *format*, Blocksize: *size***

Explanation

This message displays the record format that will be used when a new data set is created. It also displays the block size that will be used.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2531I **Record format: *format*.**

Explanation

This message displays the record format that will be used when a new data set is created.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2532I **Record format: *format*, Lrecl: X**

Explanation

This message displays the record format that will be used when a new data set is created. It also displays the logical record length X that indicates that records may exceed 32768 bytes.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2533I **Record format: *format*, Lrecl: *length***

Explanation

This message displays the record format that will be used when a new data set is created. It also displays the logical record length.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2534I **Record format: *format*, Lrecl: X, Blocksize: *size***

Explanation

This message displays the record format that will be used when a new data set is created. It also displays the logical record length X that indicates that records may exceed 32768 bytes and the block size.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2535I **Record format: *format*, Lrecl: *length*, Blocksize: *size***

Explanation

This message displays the record format that will be used when a new data set is created. It also displays the logical record length and block size.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2536I ***number*** **Mbytes transferred in** ***number*** **seconds. Transfer rate** ***value*** **Kbytes/sec.**

Explanation

This message indicates the number of bytes transferred, and the rate of transfer for a data transfer to or from the other host. If the number of bytes transferred is less than four Gigabytes, message EZA1617I is displayed.

Note: For smaller files (less than 360 KB), the reported time (and calculated rate) value reflects the amount of time to transfer the file to the TCP transport layer more than it reflects the network rates.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2537E **File transfer failed. File contains records that are longer than the LRECL of the new file.**

Explanation

Options NOTRUNCATE and NOWRAPRECORD are set. The file being transferred contains records that are longer than the LRECL of the new file.

System action

The transfer is ended. FTP continues.

Operator response

Increase the value of LRECL for the new file or change the LOCSITE options.

System programmer response

No action is needed.

Module

EZAFTPCG

Procedure name

rcvfile()

EZA2541I **Wildcards not allowed in PDS name**

Explanation

The command was issued requesting members of a partitioned data. 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 command is rejected.

Operator response

Correct the data set name and reissue the request.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2542I ***command* fails: *path name*. User not authorized**

Explanation

The command was issued for 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 command is rejected.

Operator response

Contact the owner of the data set for authorization to read the data set.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2543I **Error recalling data set *data_set***

Explanation

The command requested members of a partitioned data set. The PDS was migrated and needed to be recalled to read the PDS directory. FTP attempted to recall the data set, but was unable to successfully recall the data set.

System action

The command is rejected.

Operator response

Reissue the command. If the problem persists, contact the system programmer.

System programmer response

Determine why the data set cannot be recalled and correct the problem.

Module

EZAFTPMR

Module

EZAFTPMR

EZA2544I **Data set *dsname* is migrated and NoAutoRecall is specified.**

Explanation

A command was issued to process a data set, but the data set is migrated and FTP is currently in NoAutoRecall mode.

System action

The command is rejected.

Operator response

Issue the LOCSITE AUTORECALL command to allow the data set to be recalled, and then reissue the FTP command.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2545I **Error mounting volume containing data set *dsname***

Explanation

The volume containing the requested data set was not mounted to the FTP system. FTP attempted to have the volume mounted, but was unable to get the volume mounted.

System action

The command is rejected.

Operator response

Reissue the command. If the problem persists, contact the system programmer.

System programmer response

Determine why the requested volume cannot be mounted to the MVS system.

Module

EZAFTPMR

EZA2546I **Volume containing *dsname* is not mounted and NoAutoMount specified.**

Explanation

The command requested MVS data set *dsname*, but the volume containing *dsname* was not mounted to the system and FTP was currently in No Automount mode.

System action

The command is rejected.

Operator response

If possible, issue the LOCSITE 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

No action is needed.

Module

EZAFTPMR

EZA2547I **Partitioned data set *'path name'* does not exist.**

Explanation

The command requested that members of a partitioned data set be listed, but the requested data set could not be found.

System action

The command is rejected.

Operator response

Verify that the PDS name was entered correctly.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2548I **‘*data_set(member)*’ requests members but *data_set* is not a partitioned data set.**

Explanation

The 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 command is rejected.

Operator response

Verify that the correct data set name was entered. Reissue the command with the correct data set name.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2549I **Error opening *data_set_name***

Explanation

The command requested members of a PDS, but FTP 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 command is rejected.

Operator response

Reissue the command. If the problem persists, contact the system programmer.

System programmer response

Correct the problem with the data set.

Module

EZAFTPMR

EZA2550W **Name length error for filename *name***

Explanation

The pathname provided for a data transfer (get, put, append), 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 file name is 255 -- the total pathname maximum length is 1023.

System action

The request is not performed.

Operator response

Reissue the request with a name that meets the limits for the type of data set or file.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2551W **Unknown HOME directory for filename *name***

Explanation

A command was issued that required the client to resolve the user's HOME directory name (for example, a file name was entered that began with the directory notation ~/) but the client was unable to determine the user's HOME directory.

System action

The command is rejected. FTP continues.

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

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2552W **Nonexistent GDG data set *dsname***

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.

Operator response

Reissue the command with a valid relative number for the data set.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2553W **Unable to send *dsname***

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.

Operator response

Ensure that the GDG exists.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2554W ***cmd* fails: *filename* is a directory.**

Explanation

The file that is named is a directory. The get, put, and append commands do not support file transfer for a directory file.

System action

The command is rejected. FTP continues.

Operator response

Reissue the command specifying a name of a file that is not a directory.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2555W ***cmd* fails: *filename* is a pipe or FIFO.**

Explanation

The file that is named is a pipe or FIFO special file. The file transfer subcommands do not support files of this type.

System action

The command is rejected. FTP continues.

Operator response

Reissue the command specifying a name of a file that is not a pipe or FIFO special file.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2556W ***cmd*** **fails:** ***filename*** **is a character special file.**

Explanation

The file that is named is a character special file. The file transfer subcommands do not support files of this type.

System action

The command is rejected. FTP continues.

Operator response

Reissue the command specifying a name of a file that is not a character special file.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2557W ***cmd*** **fails:** ***filename*** **is an unknown type.**

Explanation

The file that is named is an unknown type. The file transfer subcommands do not support files of this type.

System action

The command is rejected. FTP continues.

Operator response

Reissue the command specifying a name of a file that is not an unknown file type.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2558W ***cmd* fails: *filename* does not exist.**

Explanation

The file that is named does not exist. The put and append subcommands require that the file exists.

System action

The command is rejected. FTP continues.

Operator response

Reissue the command specifying a name of a file exists.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2560W **Volume is not ready and automatic tape mounts are not allowed.**

Explanation

A request to read from a data set or write to a data set on a tape volume was received. The tape volume is not mounted and you have requested NoAutoTapeMount.

System action

The request is not performed.

Operator response

Request that the tape be mounted by the operator and then reissue the request. Also, you can code the FTP.DATA statement AUTOTAPEMOUNT to allow automatic tape mounts to occur.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2561W **Error allocating tape data set *dsname***

Explanation

The client attempted to allocate to a data set on a tape volume. The dynamic allocation was unsuccessful.

System action

The command is rejected. FTP continues.

Operator response

No action is needed.

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.

Module

EZAFTPCG, EZAFTPCP

EZA2562W **Allocation of *dsn* failed (error code *s99error* info code *s99info* S99ERSN *s99ersn*)**

Explanation

The client attempted to allocate to a data set on a DASD or tape volume. The dynamic allocation was unsuccessful.

In the message text:

dsn
The data set name.

s99error
The hexadecimal Error Reason Code from an SVC 99 request or a DYNALLOC request. See the information about interpreting Error Reason Codes from DYNALLOC in z/OS MVS Programming: Authorized Assembler Services Guide.

s99info
The hexadecimal Information Reason Code from an SVC 99 request. See the information about interpreting Information Reason Codes from DYNALLOC in z/OS MVS Programming: Authorized Assembler Services Guide.

s99ersn
The hexadecimal SMS Reason Code from SVC 99 request. See the information about S99RBX fields in z/OS MVS Programming: Authorized Assembler Services Guide for more information about SMS Reason Codes.

System action

The command is rejected. FTP continues.

Operator response

No action is needed.

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.

User response

Contact the system programmer.

Problem determination

Look up the S99ERROR, S99INFO, and S99ERSN codes information as described in the explanation and take the actions recommended there. If the FTP client trace was enabled at the time of failure, and at least option FSC was enabled, you might be able to obtain more information by inspecting trace output for messages with the tag *alloc_dasd*. See the information about diagnosing FTP client problems with tracing in z/OS Communications Server: IP Diagnosis Guide for more information about using traces.

Source

z/OS Communications Server TCP/IP: FTP

Module

ezaftpcc, ezaftpcg, ezaftpcp

Routing code

*

Descriptor code

*

Automation

Not applicable.

Example

Not applicable.

EZA2563W **Data set *dsname* used exclusively by someone else.**

Explanation

A data transfer (get, put or append) of a partitioned data set (PDS) was requested and the member requested is currently in use.

System action

The operation is not performed.

Operator response

Try the operation later when no one else is using the member of the PDS.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2564W **Open of *dsname* failed.**

Explanation

The data set cannot be opened.

System action

The requested operation is terminated. FTP continues.

Operator response

Rerun the FTP with tracing turned on. Contact the system programmer.

System programmer response

Look at the trace and find one of the following trace entry sequences:

-
  ```
  seq_open_file: xyz  -> ...
  seq_open_file: failed (aa): bbbb
  ```

  The first line describes open mode with values *x*, *y*, and *z* that are defined as follows:

  ```
  x = I --> Input
  x = O --> Output

  y = S --> Stream I/O
  y = R --> Record I/O

  z = B --> Binary stream
  z = T --> Text stream
  ```

  The second line describes the error that occurred. *aa* is the errno value returned from the fopen of file. *bbbb* is the text associated with the errno value. See the following values and text that could be observed:

  (61): Error trying to define file
  The file is already opened.

-
  ```
  seq_open_file: Using BSAM, I, SYS00006, , 29
     0, 0, 0
  seq_open_file: BSMOP returned NULL, rc xxx, reason yyy
  ```

  The second line describes the error which occurred on the open. Use the return code (rc) and reason code to determine why the error occurred.

  For return code 80a, reason code 10, or return code 4, FTP was unable to obtain storage to open the data set. Increase the region size for the FTP job or user.

Module

EZAFTPCG, EZAFTPCP

EZA2565W **Command fails for *dsname*. User not authorized.**

Explanation

The 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.

Operator response

Contact the owner of the data set for authorization to read the data set.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2566E **Requested action not taken: invalid REST parameter.**

Explanation

A command is being processed as part of a restart of a checkpointed command. Before the command was sent, a REST command was processed and a restart marker was saved. The restart marker is incorrect.

System action

The command is rejected.

Operator response

No action is needed.

System programmer response

Look at the trace to determine the error that occurred when the repositioning was done.

Module

EZAFTPCG, EZAFTPCP

EZA2567E **Record structure requires stream mode (mode S)**

Explanation

Record structure transfers require the stream transmission mode (mode S). No other transmission mode is supported for record structure file transfers.

System action

No files are transferred.

Operator response

Use the STREAM command or MODE S command to change the transmission mode to stream mode. For more information about these commands, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2568E **Transfer aborted: attempt to connect to *db2name* failed *(sql_rc)***

Explanation

FTP attempted to process a SQL request, but was unable to connect to the Db2 subsystem named *db2name*.

(sql_rc) is the SQL return code. See http://www.ibm.com/support/knowledgecenter/SSEPH2/welcome for a complete description of SQL return codes.

System action

No data is sent. FTP continues.

Operator response

If the Db2 subsystem name *(db2name)* is incorrect, issue a 'locsite db2=' command for the correct Db2 subsystem name, and resubmit the SQL query. If the Db2 subsystem name is correct, contact the system programmer for FTP.

System programmer response

Ensure that the Db2 subsystem has been started.

Module

EZAFTPCP

EZA2569E **Transfer aborted: SQL not available. Attempt to open plan *plan_name* failed *(sql_rc)***

Explanation

FTP attempted to process a SQL request, but was unable to open the FTP Db2 subsystem plan named *plan_name*

(sql_rc) is the SQL return code. See http://www.ibm.com/support/knowledgecenter/SSEPH2/welcome for a complete description of SQL return codes.

System action

No data is sent. FTP continues.

Operator response

If FTP SQL support should be available, contact the system programmer for FTP. If a SQL query was not intended, issue **locsite filetype=** command to change the file type from **SQL** to **SEQ** before the next FTP put command.

System programmer response

Ensure that a BIND was done for plan *plan_name* and that execute authorization was granted for the plan.

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 client is to use.

Module

EZAFTPCP

EZA2570E **Transfer aborted: SQL PREPARE/DESCRIBE failure**

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.

Operator response

Correct the error indicated in the output file, and resubmit the SQL request.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2571E **Transfer aborted: SQL CURSOR DECLARE failure**

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.

Operator response

If possible, correct the error indicated in the output file and resubmit the SQL request. Otherwise, contact the system programmer for FTP.

System programmer response

Correct the error indicated by the message contained in the output file. For a detailed explanation of Db2 error codes, see http://www.ibm.com/support/knowledgecenter/SSEPH2/welcome.

Module

EZAFTPCP

EZA2572E **Transfer aborted: SQL CURSOR OPEN failure**

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.

Operator response

If possible, correct the error indicated in the output file and resubmit the SQL request. Otherwise, contact the 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 http://www.ibm.com/support/knowledgecenter/SSEPH2/welcome.

Module

EZAFTPCP

EZA2573E **Transfer aborted: unsupported SQL statement**

Explanation

FTP retrieved a file (or data set) while the local file type was SQL, but the file contains an SQL statement that is not supported by FTP.

System action

The PUT command is rejected.

Operator response

If an SQL query was intended, issue the PUT command for a file that contains an SQL SELECT statement. If an SQL query was not intended, issue a **site filetype=** command to change the current setting of filetype.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2574E **Transfer aborted: SQL FETCH error *(sql_rc)***

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.

(sql_rc) is the SQL return code. See http://www.ibm.com/support/knowledgecenter/SSEPH2/welcome for a complete description of SQL return codes.

System action

FTP continues.

Operator response

Resubmit the query. If the problem persists, contact the system programmer for FTP.

System programmer response

Correct the problem described in *(sql_rc)*. See http://www.ibm.com/support/knowledgecenter/SSEPH2/welcome for a complete description of SQL return codes.

Module

EZAFTPCP

EZA2575E **Transfer aborted: insufficient storage to process SQL request**

Explanation

A SQL query was being attempted through FTP, but there is not enough system storage available for FTP to complete the request.

System action

No data is sent. FTP continues.

Operator response

Try again. If the problem persists, contact the system programmer for the FTP client.

System programmer response

Determine why FTP is unable to acquire dynamic storage and correct the problem.

Module

EZAFTPCP

EZA2576E **Transfer aborted: internal error while processing SQL request**

Explanation

FTP has encountered an internal error while processing a SQL request. No data is sent.

System action

FTP continues.

Operator response

Try again. If the problem persists, contact the system programmer for FTP.

System programmer response

If the problem occurs repeatedly, use the client's debug facility to get a trace of the FTP operation and contact the IBM Support Center.

Module

EZAFTPCP

EZA2578I **Local directory name set to hierarchical file *directory***

Explanation

The name of the local directory is a hierarchical file directory. Files are grouped in a directory, which is a special kind of file consisting of the names of a set of files and other information about them.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Module

EZAFTPCL

EZA2580E ***cmd*** **cmd failed :** ***error_description***

Explanation

The FTP user issued the command *cmd*. While processing the command, the FTP client issued a C runtime library function that did not complete successfully.

error_description describes error.

System action

The command *cmd* is not executed.

Operator response

Reissue the command. If the problem persists, contact the system programmer with the error message.

System programmer response

Correct the error indicated by *error_description*.

Module

EZAFTPCD

EZA2581I **HFS directory** ***directory*** **is the current working directory**

Explanation

The user has issued the LCD command to change the current working directory. The LCD command has completed successfully and has changed the current working directory to the z/OS UNIX file system directory indicated by the *directory* value.

System action

The current working directory is changed to the new directory.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2582E ***cmd*** **failed: Unknown HOME directory**

Explanation

A command was issued that required the client to resolve the user's HOME directory name (for example, a pathname was entered that began with the directory notation ~/) but the client was unable to determine the user's HOME directory.

System action

The command is rejected. FTP continues.

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

No action is needed.

Module

EZAFTPCD

EZA2583I **Working Directory for GET is NULL Device**

Explanation

The user has entered the LCD *DEV.NULL command to change the current working directory of the client to the NULL directory. When the current working directory is the NULL directory, data transferred to the client from the server (GET or RETR), will not be stored at the client, but will be received and discarded. The NULL directory is used only for data transferred from the server to the client. Data transferred from the client to the server (PUT) will continue to use the working directory which was in effect prior to the issuance of the LCD *DEV.NULL command. The *directory* which will be used for the PUT command is described in the message that follows this one:

- EZA2584I for PUT is “*directory*” partitioned data set
- EZA2585I for PUT is the z/OS UNIX file system directory *directory*
- EZA2586I for PUT is “*directory*” name prefix.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD, EZAFTPCL

EZA2584I **for PUT is “*directory*” partitioned data set.**

Explanation

This message is part of a group. See EZA2583I for message group information.

EZA2585I **for PUT is HFS directory *directory***

Explanation

This message is part of a group. See EZA2583I for message group information.

EZA2586I **for PUT is “*directory*” name prefix.**

Explanation

This message is part of a group. See EZA2583I for message group information.

EZA2587E **LCD ../ only valid for HFS directories**

Explanation

The LCD 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 a z/OS UNIX file system directory.

System action

The LCD command is rejected.

Operator response

Reissue the LCD command with the corrected pathname.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2588E **Unable to connect to alternate TCPIP *name***

Explanation

Error returned from setibmopt routine.

System action

None.

Operator response

If the TCPIP start parameter is used, check the spelling of the parameter. If the spelling is correct, check that the stackname specified in the start parameter matches the TCPIPJOBNAME specified in the TCPIP.DATA configuration file. For more information about the TCPIP.DATA configuration file, see the z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA2589E **Connection to server interrupted or timed out. *operation***

Explanation

The FTP server is no longer responding to the FTP client or the user has caused an interrupt (for example, Cntrl-c).

operation is the operation that was in progress at the time of the interruption. Possible values are:

Initial connection
During initial connection to a non-IPv6 server.

Initial IPv6 connection
During initial connection to an IPv6 server.

Waiting for data connection
Waiting for the server to establish the data connection.

Sending a command
Sending a command to the server on the control connection.

Sending ABORT command
Sending the ABORT command to the server.

Receiving data
Receiving data on the data connection.

Sending data
Sending data on the data connection.

Waiting for reply
Waiting for a reply from the server.

Sending command to SOCKS server
Sending a command to a SOCKS server.

Waiting for reply from SOCKS server
Waiting for a reply from a SOCKS server.

Establishing data connection through SOCKS server
While trying to establish a data connection through a SOCKS server.

Initial connection to SOCKS server
During initial connection to a SOCKS server.

System action

The current subcommand is ended.

Operator response

The server may not be responding because of traffic in the network. In this case a retry may be successful. The server may have crashed, in which case it will need to be restarted before attempting communications again. A complete description of each operation's relevant client timers and recommended steps for diagnosis can be found in the section on diagnosing FTP transfer failures with EZA2589E in z/OS Communications Server: IP Diagnosis Guide. If the problem continues, contact the system programmer.

System programmer response

Possible sources of the failure include the client, server, network, firewall, and configuration errors. See the section on diagnosing FTP transfer failures with EZA2589E in z/OS Communications Server: IP Diagnosis Guide for recommended actions to diagnose the failure.

Module

EZAFTPSC

EZA2590E ***function*** **error from** ***location*** **-** ***error***

Explanation

The FTP client has issued a socket call and received an error return code.

function is the C library function that failed and returned an error.

location is the location in the FTP client.

error is the C run-time library error message for the failure. For more information, see z/OS Language Environment Runtime Messages.

System action

Communications are interrupted. The current operation is ended.

Operator response

Try the operation again. If the problem persists, contact the system programmer.

System programmer response

If the problem persists, report the contents of the message to the IBM Software Support center.

Module

EZAFTPSC

EZA2591I **Waiting for recall of "*dsname*"**

Explanation

You issued a subcommand which referenced the data set *dsname*, which has been migrated.

System action

The client attempts to recall the data set.

Operator response

No action is needed.

System programmer response

If you do not want the FTP client to recall migrated data sets, use the **LOCSIte** subcommand to set the autorecall mode to noautorecall. See z/OS Communications Server: IP User's Guide and Commands for information about the **LOCSIte** subcommand.

Module

EZAFTPCD

EZA2592I **Waiting for mount for "*dsname*"**

Explanation

You issued a subcommand which references *dsname*, an existing data set which is cataloged on a volume which is not mounted. The client is attempting to mount the volume so it can complete the request.

System action

The client waits for the volume to be mounted.

Operator response

If you do not want the FTP client to mount unmounted volumes, use the **LOCSIte** subcommand to set automount mode to noautomount. See z/OS Communications Server: IP User's Guide and Commands for information about the **LOCSIte** subcommand.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2593I **HFS directory “*directory*” created.**

Explanation

The indicated the z/OS UNIX file system directory has been created.

System action

The directory is created. FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2594E **LMKDIR failed. Unit=*unit_name* invalid for PDS.**

Explanation

The LMKDIR command was issued to create a new PDS, but the current setting of the LOCSITE UNIT parameter was not valid for a PDS (for example, the LOCSITE UNIT was set to TAPE and a PDS cannot be allocated on a tape).

System action

The LMKDIR command is rejected.

Operator response

Issue the LOCSITE 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 LMKDIR command.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2595E **LMKDIR failed. DCBDsn data set *dcbdsn_name* has a recfm of *recfm* which is invalid for a PDS.**

Explanation

The LMKDIR command was issued to create a new PDS. The LOCSITE DCBDsn parameter 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 LMKDIR command is rejected.

Operator response

Use the LOCSITE 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 use a combination of DCBDsn and RECFM LOCSITE 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 LOCSITE parameters, reissue the LMKDIR command.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2596E **LMKDIR fails: *new_directory*. User not authorized**

Explanation

The LMKDIR 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 LMKDIR command is rejected.

Operator response

Contact the security administrator to get the necessary authorization for creating the data set.

System programmer response

No action is needed.

EZA2597E **LMKDIR failed. Volume=*serial* is a tape and is invalid for a PDS.**

Explanation

The LMKDIR command was issued to create a new PDS. The LOCSITE 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 LMKDIR command is rejected.

Operator response

Issue the LOCSITE VOLUME command to change or reset the volume serial for new data sets to a direct access volume, then reissue the LMKDIR command.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2598E **LMKDIR failed. Recfm *recfm* is invalid for a PDS.**

Explanation

The LMKDIR command was issued to create a new PDS. The LOCSITE RECFM parameter had a value of *recfm*, which is not a valid record format for a PDS.

System action

The LMKDIR command is rejected.

Operator response

Issue the LOCSITE RECFM command to change the setting of RECFM to a record format that is valid for a PDS, then reissue the LMKDIR command. Incorrect record formats are FBS, VBS, and VS).

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2599W ***ftp_data* file, line *line_number*: NCP obsolete: parameter not required.**

Explanation

While processing the FTP.DATA file, the FTP client encountered the NCP keyword. This keyword is no longer required. Access method buffer information is computed internally.

Operator response

No action is needed.

System programmer response

If the FTP.DATA file is used only by the FTP server and client for z/OS UNIX System Services MVS, and is not shared with another server or client that needs the keyword, remove the keyword from the FTP.DATA file. See the z/OS Communications Server: IP Configuration Reference for information on the parameters of the FTP.DATA file.

Module

EZAFTPEP

EZA2600E **Command *command name* is not a valid proxy command.**

Explanation

The indicated command was not a valid command that can be sent to the proxy server. For example, local client commands cannot be sent to the proxy server.

System action

The command is rejected.

Operator response

Reissue the command without the proxy prefix, or enter a valid command for the proxy server.

System programmer response

No action is needed.

Module

EZAFTPPC

EZA2601W **NOWRTAPEFastio already specified. WRTAPEFastio ignored.**

Explanation

The NOWRTAPEFastio parameter was already specified on this LOCSITE subcommand. The WRTAPEFastio parameter is ignored.

System action

FTP Continues

Operator response

Issue the LOCSITE subcommand again with the required setting.

System programmer response

No action is needed.

EZA2602E **Error reading the file**

Explanation

The local file could not be read.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2603E **Error sending the file**

Explanation

The local file could not be sent.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2604I **DSORG=DA requires structure: FILE, mode: STREAM, and transfer type: IMAGE.**

Explanation

A command was issued that required the server to transfer an MVS data set which had a dsorg of DA (Direct Access). Data sets with a dsorg of DA can only be transferred in stream mode with the transfer type set to image (binary).

System action

The command is rejected. FTP continues.

Operator response

Set the structure to FILE, the mode to STREAM, the transfer type to IMAGE, and reissue the command.

System programmer response

No action is needed.

EZA2605W **Local data set is full (*value*)**

Explanation

FTP was unable to write to the local data set because it is full. One of the following explicit values was received when the write failed:

19
An unrecoverable error has permanently marked the data set in error.

33
An attempt was made to extend a non-extendable file.

65
A write system error occurred.

System action

FTP continues.

Operator response

Request a larger space allocation for the local data set and try the transfer again.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2606W **File I/O error *value***

Explanation

FTP experienced a file I/O error when reading from or writing to a data set or file.

value is an unrecognized error value from the file I/O request.

System action

FTP continues.

Operator response

Notify the system programmer about the error.

System programmer response

Activate the debug option and reenter the request. Contact the IBM Software Support Center.

Module

EZAFTPCG

EZA2607W **Transfer aborted due to receive error (*file_status*)**

Explanation

The transfer of data ended because of an error receiving data from the data connection.

file_status is an FTP error number. The message immediately following this one gives the complete description of the error.

System action

FTP continues.

Operator response

Notify the system programmer about the error.

System programmer response

Correct the error described in *file_status*.

Module

EZAFTPCG

EZA2617W ***parameter2* already specified. *parameter1* ignored.**

Explanation

The LOCSITE subcommand parameters *parameter1* and *parameter2* are mutually exclusive and cannot be specified on the same LOCSITE subcommand. *parameter2* has already been encountered on this LOCSITE subcommand, causing *parameter1* to be rejected.

System action

parameter2 is used instead of *parameter1*. FTP continues.

Operator response

If you want to use *parameter1*, issue a LOCSITE subcommand specifying *parameter1*, but not *parameter2*. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information about FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information about the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2624W **WRTAPEFastio already specified. NOWRTAPEFastio ignored.**

Explanation

The WRTAPEFastio parameter was already specified on this LOCSITE subcommand. The NOWRTAPEFastio parameter is ignored.

System action

FTP continues.

Operator response

Issue the LOCSITE subcommand again with the required setting.

System programmer response

No action is needed.

EZA2625W **Invalid format data set name “*name*”. Dcbdsn parameter ignored.**

Explanation

The data set name specified for the DCbdsn parameter of the LOCSITE 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.

Operator response

Reissue the LOCSITE command with a valid MVS data set name. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

Assist the user as necessary.

EZA2627I **Internal debug option *value* enabled**

Explanation

The internal debug options are set on. If *value* is 1, regular tracing is on. If *value* is 2, detailed tracing is on.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2628W ***parm* parameter's value *(value)* must be between *low value* and *high value*. The parameter is ignored.**

Explanation

The LOCSITE command was specified with the *parm* parameter, but the value, *value*, specified for the *parm* parameter was outside the valid range.

System action

The *parm* parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guidefor information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2631W	**CHKptint parameter *(value)* is larger than a 4 byte integer. Chkptint ignored.**

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.

Operator response

Reissue the LOCSITE command with a valid value specified for the CHKptint parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guidefor information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2632I	**Data set name too long. Use MVS or HFS naming conventions.**

Explanation

A command was entered with a data set name that was longer than the MVS maximum of 44 characters or the z/OS UNIX file system maximum of 1023 characters.

System action

The command is rejected.

Operator response

Reissue the command with the correct data set or z/OS UNIX file name.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2633I **File name too long. Use HFS naming conventions.**

Explanation

The command was entered with a file name that was longer than the z/OS UNIX file system maximum of 1023 characters.

System action

The command is rejected.

Operator response

Reissue the command with the correct z/OS UNIX file name.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2634I ***cmd* failed: Unknown HOME directory**

Explanation

A command was issued that required FTP to resolve the user's HOME directory name (for example, a pathname was entered that began with the directory notation ~/) but FTP was unable to determine the user's HOME directory.

System action

The command is rejected. FTP continues.

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

No action is needed.

Module

EZAFTPMR

EZA2635I **Member name too long. Use MVS naming conventions.**

Explanation

The 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 command is rejected.

Operator response

Reissue the command with the correct member name syntax.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2636E **TSO *Command* failed with return code *value***

Explanation

The TSO command requested completed in error.

System action

The command fails.

Operator response

Correct the command and reissue. If the command is correct but still fails, turn on debug and reissue the command. Give the debug output to the system programmer.

System programmer response

The debug output from the TSO command shows the return code and reason code from the IKJEFTSR service facility routine, which is used to invoke the TSO command from FTP. See z/OS TSO/E Programming Services for a list of return codes and reason codes for the IKJEFTSR service facility routine.

Module

EZAFTPCU

EZA2637I **No data sets found**

Explanation

The command requested a listing of one or more data sets, but FTP could not find any data sets that matched the requested pathname.

System action

The command is rejected.

Operator response

Verify that the correct data set name or path was entered.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2638I **error reading PDS directory**

Explanation

The command requested members of a partitioned data set, but FTP was unable to read the directory of the PDS.

System action

The command is rejected.

Operator response

Reissue the command. If the problem persists, contact the system programmer.

System programmer response

Correct the problem with the data set.

Module

EZAFTPMR

EZA2639I **Qualifier too long. Use MVS naming conventions.**

Explanation

A 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 command is rejected.

Operator response

Correct the data set name and reissue the command.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2640I **Error allocating storage for list**

Explanation

FTP was not able to allocate the storage necessary to process the command.

System action

The command is rejected.

Operator response

Reissue the command. If the problem persists, contact the system programmer.

System programmer response

FTP might need to be started with a larger region size.

Module

EZAFTPMR

EZA2641I **Patterns (% and *) may only be specified in the last qualifier when in directory mode**

Explanation

The command was entered with a pathname that contained the global file-name characters % and *. The global file-name characters were not in the last qualifier of the pathname, and the MVS server was currently in directory mode. The global file-name characters can only be used in the last qualifier when in directory mode.

System action

The command is rejected.

Operator response

Either correct the pathname to have the global file-name characters in the last qualifier and reissue the command in directory mode, or issue the LOCSITE/SITE DATASETMODE command to change the server to data set mode and then reissue the command as is.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2643I **Unable to obtain data set list**

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 command is rejected.

Operator response

Try the command again. If the problem persists, contact the system programmer.

System programmer response

If necessary, re-create the problem with FTP traces turned on. Locate the last occurrence of the trace message `pslist: reason code rc from CSI`. This trace message will contain the reason code from the service call that was unsuccessful. Determine the cause and correct the problem.

Recommend Trace: UTL

Module

EZAFTPMR

EZA2644I **No members found.**

Explanation

The command requested members of a partitioned data set, but no members were found in the PDS that matched the requested name.

System action

The command is rejected.

Operator response

Verify that the member name or path was entered correctly.

System programmer response

No action is needed.

Module

EZAFTPMR

EZA2658W ***parm* parameter *(value)* is not valid. *parm* ignored.**

Explanation

The LOCSITE command was entered with the *parm* parameter, but the value specified, *value*, was not a valid value. Valid values for the *parm* parameter are Catlg and Delete.

System action

The *parm* parameter is ignored.

Operator response

Reenter the LOCSITE subcommand, specifying the correct value for *parm*. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2659W ***parm* parameter *(value)* is not numeric. *parm* ignored.**

Explanation

The LOCSITE command was specified with the *parm* parameter, but the value, *value*, specified for the *parm* parameter was not a numeric value.

System action

The *parm* parameter is ignored.

Operator response

Reissue the command with a valid value. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2660I **Workarea allocation failed.**

Explanation

The allocation of the workarea for qdisk failed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2661W **Restart requires file structure**

Explanation

A RESTART command was issued, but the structure is not file structure.

System action

FTP continues.

Operator response

Change the structure to file (issue file or structure f) and reissue the RESTART command.

System programmer response

No action is needed.

Module

EZAFTPCB

EZA2662W **NCP parameter obsolete: Parameter ignored.**

Explanation

The NCP parameter is obsolete. It is no longer required because values are computed internally.

System action

FTP continues.

Operator response

None required, the parameter is ignored.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2663W ***parameter* not allowed when user not logged in. *parameter* ignored.**

Explanation

The LOCSITE subcommand was issued with the *parameter*, but the user was not currently logged on to the server with a valid user ID. The user must first log in with a valid user ID and password before issuing the LOCSITE subcommand with the *parameter* parameter.

System action

FTP continues.

Operator response

Log in to the server using the USER and PASS subcommands, then reissue the LOCSITE subcommand with the *parameter* parameter.

System programmer response

Module

EZAFTPCK

EZA2664W **CHMod invalid syntax. Specify CHMod mode filename. CHMod ignored.**

Explanation

The CHMod parameter of the LOCSITE command was entered incorrectly. The correct syntax of the CHmod parameter of the LOCSITE subcommand is LOCSIte 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.

Operator response

Reissue the LOCSITE command with the correct syntax for the CHmod parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

EZA2665W **CHMod invalid filename *filename***

Explanation

The file name specified by the CHMod parameter of the LOCSITE subcommand was longer than the z/OS UNIX file name maximum of 1023 characters. If the file name operand of the CHMod parameter began with a forward slash (/), then the *filename* value is the file name operand used “as is”; otherwise, the *filename* value is the file name operand appended to the current working directory.

filename is the name of the file that is not valid.

System action

The CHMod parameter is ignored.

Operator response

Reissue the LOCSITE command with a valid value specified for the file name on the CHMOD parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2667W **CHMod filename is not a valid HFS file. CHMod ignored**

Explanation

The file name specified on the CHMod parameter of the LOCSITE subcommand was not an z/OS UNIX file. The CHMod parameter is valid only for z/OS UNIX files.

System action

The CHMod parameter is ignored.

Operator response

Reissue the LOCSITE command with a valid value specified for the file name on the CHMod parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2668W **CHMod cannot access file *filename* : *error_description***

Explanation

The FTP client encountered an error while attempting to obtain the file information for *filename*.

error_description describes the C runtime library error that was encountered while accessing *filename*.

System action

The CHMod parameter is ignored.

Operator response

Reissue the LOCSITE command with a valid value specified for the file name on the CHMod parameter. If the problem persists, contact the system programmer.

System programmer response

If necessary, correct the error indicated by *error_description*.

Module

EZAFTPCK

EZA2669W **CHMod failed: *error_description***

Explanation

The FTP client encountered an error while attempting to change the permission bits of the file.

error_description describes the C runtime library error that was returned in response to the chmod() request.

System action

The CHMod parameter is ignored.

Operator response

Contact the system programmer.

System programmer response

Correct the error indicated by *error_description*

Module

EZAFTPCK

EZA2670W 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

Explanation

The LOCSITE 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.

Operator response

Reenter the LOCSITE subcommand, specifying the correct value for the mode operand. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2671W CHMod parameters missing. Specify CHMod mode filename. CHMod ignored

Explanation

The LOCSITE subcommand 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 LOCSITE CHMod *mode filename*.

System action

The CHMod parameter is ignored.

Operator response

Reenter the LOCSITE subcommand, specifying the correct operands. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2672W **UMask invalid syntax. Specify UMask umask_value. UMask ignored**

Explanation

The LOCSITE command was issued with the UMask parameter, but the syntax of the UMask parameter was incorrect. The correct syntax is LOCSITE 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.

Operator response

Reissue the LOCSITE command with a valid value for the UMask parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

Assist the user as necessary.

Module

EZAFTPCK

EZA2673I **parameter *parameter* not allowed when the control connection is not secured**

Explanation

You entered an FTP client subcommand during a secure session while the control connection was not secured. The subcommand parameter is not allowed while the control connection is not secured.

In the message text:

parameter

The subcommand parameter that is not allowed when the control connection is not secured.

If the subcommand parameter is TLSRFCLEVEL=DRAFT, the user is trying to transition the client to the Internet draft level of RFC 4217, *On Securing FTP with TLS*. However, the Internet draft level of RFC 4217 implemented by z/OS FTP does not support a TLS-secured session with an unsecured control connection.

System action

FTP ignores the subcommand parameter.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

If you are trying to switch the client from using TLSRFCLEVEL=RFC4217 to using TLSRFCLEVEL=DRAFT, use the AUth subcommand with the TLS parameter to secure the control connection, and try switching the client again. See the information about the AUth subcommand information in z/OS Communications Server: IP User's Guide and Commands.

Problem determination

No action is needed.

Source

z/OS Communications Server TCP/IP: FTP

Module

Not applicable.

Routing code

10

Descriptor code

12

Example

```
Command:  ccc
>>>CCC
200 CCC command successful
locsite TLSRFCLEVEL=DRAFT NOFWFRIENDLY
EZA2673I parameter TLSRFCLEVEL=DRAFT is not allowed when control connection is  not secured
```

EZA2674W UMask value invalid. Specify UMask value as 1 - 3 octal digits

Explanation

The LOCSITE 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.

Operator response

Reissue the LOCSITE command with a valid value for the UMask parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

Assist the user as necessary.

Module

EZAFTPCK

EZA2675W **200-UMask value missing. Specify UMask value as 1 - 3 octal digits**

Explanation

The LOCSITE 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.

Operator response

Reissue the LOCSITE command with a valid value for the UMask parameter. See the z/OS Communications Server: IP User's Guide and Commands or z/OS Communications Server: IP Configuration Guide for information on FTP in general and to z/OS Communications Server: IP User's Guide and Commands for information on the parameters of the LOCSITE subcommand.

System programmer response

Assist the user as necessary.

Module

EZAFTPCK

EZA2676I **Connected to:*host*, Port:FTP control (21), logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. The host is the remote host to which you are connected. The port is the default FTP control port, 21, on the remote host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2677I **Connected to:*host*, Port:FTP control (21), not logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. The host is the remote host to which you are connected. The port is the default FTP control port, 21, on the remote host.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2689I **Prompting: *prompt*, Globbing: *glob***

Explanation

This message is issued in response to the LOCSTAT subcommand. The prompt value indicates whether the PROMPT setting is on or off. On indicates that interactive prompting on multiple commands is on. The glob value indicates whether the GLOB setting is on or off. On indicates that character expansion of local file names is on.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2690I **Invalid response to prompt. Try again.**

Explanation

You entered an invalid response to an interactive prompt.

System action

FTP continues.

Operator response

Enter one of the valid responses that are displayed in the prompt message.

System programmer response

No action is needed.

Module

EZAFTPCA

EZA2691I **Globbing on**

Explanation

File-name parameters for the multiple commands (mdelete, mget, and mput) are expanded. That is, the asterisk is used as a wildcard.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2692I **Globbing off**

Explanation

File name parameters for the multiple subcommands (mdelete, mget, and mput) are not expanded. That is, the asterisk (*) is not used as a wildcard.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2693I **Interactive mode is on**

Explanation

FTP subcommands will prompt for verification before deleting, retrieving, or sending multiple files during the mdelete, mget, and mput subcommands. One of the following four values must be entered for each prompt:

Y (Yes)
execute the subcommand

N (No)
do not execute the subcommand

Q (Quit)
quit processing the subcommand immediately

S (Stop prompting)
do not prompt anymore for this subcommand.

Note: Pressing the enter key with no value is not a valid response.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2694I **Interactive mode is off**

Explanation

FTP commands will not prompt for verification before deleting, retrieving, or sending multiple files during the mdelete, mget, and mput commands.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2695I **Mdelete *filename*(Yes|No|Quit|Stop prompting)?**

Explanation

The prompt is displayed for each file before it is deleted when the mdelete subcommand is being processed.

filename is the name of the file or data set that FTP is about to delete.

You must enter one of the following values before processing continues:

Y (Yes)
Delete the file or data set indicated by *filename*. FTP will prompt you with a new file name if it finds another data set or file to delete.

N (No)
Do not delete the file or data set indicated by *filename*. FTP will prompt you with a new file name if it finds another data set or file to delete.

Q (Quit)
Quit processing the subcommand immediately (do not delete any more files or data sets).

S (Stop prompting)
Delete *filename*. Continue deleting data sets or files, but stop prompting before each file or data set deletion.

Notes:

1. Only the first character of the entered value is interrogated. That is, entering **yyyyy** has the same effect as entering **yes**.
2. Pressing the enter key with no value is not a valid response.
3. If you do not want to see the prompt the next time you issue the **mdelete** subcommand, use the **prompt** subcommand to toggle the interactive mode to off.

System action

FTP waits for your reply. The subsequent action depends on how you reply to the prompt.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCA

EZA2696I **Mget *filename*(Yes|No|Quit|Stop prompting)?**

Explanation

The prompt is displayed for each file before it is retrieved when the mget subcommand is being processed.

Filename is the name of the file or data set FTP is about to retrieve.

You must enter one of the following values before processing continues:

Y (Yes)
Retrieve *filename*. FTP will prompt you with another *filename* if it finds another file or data set to retrieve.

N (No)
Do not retrieve *filename*. FTP will prompt you with another *filename* if it finds another file or data set to retrieve.

Q (Quit)
Quit processing the subcommand immediately (do not retrieve any more files or data sets).

S (Stop prompting)
Retrieve *filename*. Continue retrieving files or data sets, but do not prompt before retrieving each file.

Notes:

1. Only the first character of the entered value is interrogated. That is, entering **yyyyy** has the same effect as entering **yes**.

2. Pressing the enter key with no value is not a valid response.
3. If you do not want to see the prompt the next time you issue the **mget** subcommand, use the **prompt** subcommand to toggle the interactive mode to off.

System action

FTP waits for your reply. The subsequent action depends on how you reply to the prompt.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCA

EZA2697I **Mput *filename*(Yes|No|Quit|Stop prompting)?**

Explanation

The prompt is displayed for each file before it is sent when the **mput** command is being processed.

filename is the name of the file or data set that FTP is about to send to the server.

You must enter one of the following values before processing continues:

Y (Yes)
Send *filename*. FTP will prompt you with another *filename* if it finds another file or data set to send.

N (No)
Do not send *filename*. FTP will prompt you with another *filename* if it finds another file or data set to send.

Q (Quit)
Quit processing the subcommand immediately (do not send any more files or data sets).

S (Stop prompting)
Send *filename*. Continue sending files or data sets to the server, but do not prompt before sending each file.

Notes:

1. Only the first character of the entered value is interrogated. That is, entering **yyyyy** has the same effect as entering **yes**.
2. Pressing the enter key with no value is not a valid response.
3. If you do not want to see the prompt the next time you issue the mput subcommand, use the prompt subcommand to toggle the interactive mode to off.

System action

FTP waits for your reply. The subsequent action depends on how you reply to the prompt.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCA

EZA2698I **Proxy connection to:*host*, Port: *port*, logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. A proxy connection to a remote host exists. The host is the remote host to which you are connected. The port is the port number of the remote host. This message is only displayed if a proxy connection exists.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2699I **Proxy connection to:*host*, Port: *port*, not logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. A proxy connection to a remote host exists. The host is the remote host to which you are connected. The port is the port number of the remote host. This message is only displayed if a proxy connection exists.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2700I **Proxy connection to:*host*, Port:FTP control (21), logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. A proxy connection to a remote host exists. The host is the remote host to which you are connected. The port is the default FTP control port, 21, on the remote host. This message is only displayed if a proxy connection exists.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2701I **Connected to:*host*, Port:FTP control (21), not logged in**

Explanation

This message is issued in response to the LOCSTAT subcommand. A proxy connection to a remote host exists. The host is the remote host to which you are connected. The port is the default FTP control port, 21, on the remote host. This message is only displayed if a proxy connection exists.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2702I **Proxy Local Port: *port***

Explanation

This message is issued in response to the LOCSTAT subcommand. The indicated port is the port number of the local host for the proxy connection. This message is only displayed if a proxy connection exists.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2703I **Valid proxy commands are:**

Explanation

This message is issued in response to the HELP PROXY subcommand. Following it are a list of commands which are valid on the PROXY subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2704I **For more information about proxy commands, say 'HELP ALL PROXY'**

Explanation

This message provides information about how to request help for PROXY commands using the HELP command. This message is issued in response to a HELP PROXY request.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2705E **Command not available. Unable to load Japanese translation tables in *filename*; return code: *code***

Explanation

The FTP client 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.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the return code displayed in the message.

Module

EZAFTPNC

EZA2706E **Command not available. Unable to access Japanese translation tables.**

Explanation

The FTP client cannot open the indicated translation table.

System action

FTP continues.

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 client.

Module

EZAFTPNC

EZA2707E **Command not available. Unable to load Korean translation tables in *filename*; return code: *code***

Explanation

The FTP client 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.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the return code displayed in the message.

Module

EZAFTPNC

EZA2708E **Command not available. Unable to access Korean translation tables.**

Explanation

The FTP client cannot open the indicated translation table.

System action

FTP continues.

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 client.

Module

EZAFTPNC

EZA2709E **Command not available. Unable to load traditional Chinese translation tables in *filename*; return code: *code***

Explanation

The FTP client 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.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the return code displayed in the message.

Module

EZAFTPNC

EZA2710E **Command not available. Unable to access traditional Chinese translation tables.**

Explanation

The FTP client cannot open the indicated translation table.

System action

FTP continues.

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 client.

Module

EZAFTPNC

EZA2711E **Command not available. Unable to load simplified Chinese translation tables in *filename*; return code: *code***

Explanation

The FTP client 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.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the return code displayed in the message.

Module

EZAFTPNC

EZA2712E **Command not available. Unable to access simplified Chinese translation tables.**

Explanation

The FTP client cannot open the indicated translation table.

System action

FTP continues.

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 client.

Module

EZAFTPNC

EZA2713I **Language: *name characteristics***

Explanation

This message is issued in response to the LOCSTAT subcommand when the FTP data transfer type is 'B' (DBCS). It displays the name of the particular DBCS data type, followed by any SO/SI characteristics that were specified when this data type was specified.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2714I **Proxy Command:**

Explanation

The system is prompting you to enter a proxy command.

System action

The system waits for you to enter a proxy command.

Operator response

Enter the command you want to process.

System programmer response

No action is needed.

Module

EZAFTPPC

EZA2715I **No primary connection**

Explanation

You entered a proxy command that requires both a primary and a secondary connection. The OPEN subcommand must be issued to establish a connection to the primary FTP server.

System action

FTP continues.

Operator response

Issue the OPEN subcommand with a host name.

System programmer response

No action is needed.

Module

EZAFTPCA, EZAFTPCG, EZAFTPCP

EZA2716I **Replace option is invalid for a proxy command**

Explanation

You entered a proxy get or mget command with the replace option. The option is not supported.

System action

FTP continues.

Operator response

Issue the command without replace.

System programmer response

No action is needed.

Module

EZAFTPCA, EZAFTPCG

EZA2717E **PROXY and NOTYPE are conflicting specifications.**

Explanation

You entered a proxy command with the notype option. A proxy command is normally sent to the secondary server but the notype option indicates the corresponding type command should not be sent to the server.

System action

The command is ignored. FTP continues.

Operator response

Issue the command without notype, or not as a proxy command.

Operator response

If you intended the command for the secondary server, reissue the command without the notype option. If you intended this command for the client only, reissue it without specifying proxy.

System programmer response

No action is needed.

Module

EZAFTPCT

EZA2718I **ASA control characters transferred as C control character sequences**

Explanation

This message is issued in response to the LOCSTAT subcommand. ASA control characters in ASA files opened for text processing will be converted to C control character sequences during file transfer.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2719I **ASA control characters transferred as ASA control characters**

Explanation

This message is issued in response to the LOCSTAT subcommand. ASA control characters in ASA files opened for text processing will be transferred as ASA control characters.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2720I **New data sets catalogued if a store operation terminates abnormally**

Explanation

This message is issued in response to the LOCSTAT subcommand. New data sets will be cataloged if a store operation terminates abnormally.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2721I **New data sets deleted if a store operation terminates abnormally**

Explanation

This message is issued in response to the LOCSTAT subcommand. New data sets will be deleted if a store operation terminates abnormally.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2722I **Single quotes will override the current working directory**

Explanation

This message is issued in response to the LOCSTAT subcommand. Single quotes will override the current working directory.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2723I **Single quotes will be treated as part of the filename**

Explanation

This message is issued in response to the LOCSTAT subcommand. Single quotes will be treated as part of the file name.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2724I **UMASK value is *umask***

Explanation

This message is issued in response to the LOCSTAT subcommand. Specifies the octal umask to be used when allocating new z/OS UNIX files.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2734W **Sbdataconn invalid syntax. Specify SBDATACONN=(value1,value2) or SBDATACONN=dsn. SBDATACONN ignored.**

Explanation

A LOCSITE 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 z/OS UNIX file containing translate tables generated by the CONVXLAT utility.

System action

FTP continues.

Operator response

Resubmit the LOCSITE subcommand with corrected syntax.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2735W **Sbdataconn parameter is too long. Maximum length for code page name is *length*. SBDATACONN ignored.**

Explanation

A LOCSITE subcommand has been entered with an SBDATACONN parameter specifying an invalid code page name. The SBDATACONN parameter is ignored.

System action

FTP continues.

Operator response

See the z/OS XL C/C++ Programming Guide for information on supported code set converters and valid code set names. Resubmit the corrected LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2736W **No conversion available between *parm1* and *parm2*. SBDATACONN ignored.**

Explanation

A LOCSITE 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.

Operator response

See the z/OS XL C/C++ Programming Guide for information on supported code set converters and valid code set names. Resubmit the corrected LOCSITE subcommand, with the EBCDIC code set name as the first SBDATACONN value, followed by the ASCII code set name.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2737W **Unable to set up conversion between *parm1* and *parm2*. SBDATACONN ignored.**

Explanation

A LOCSITE 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.

Operator response

Ensure that the requested code set names are for single-byte code pages. SBDATACONN is not supported for double-byte. See the z/OS XL C/C++ Programming Guide for information on supported code set converters and valid code set names.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2738W **Translate file '*file*' not found. SBDATACONN ignored.**

Explanation

A LOCSITE subcommand has been entered with an SBDATACONN parameter that specified a file name, but FTP was unable open *file*. The SBDATACONN parameter is ignored.

System action

FTP continues.

Operator response

Ensure that the file name specified on the SBDATACONN parameter is a fully qualified name of an existing MVS data set or z/OS UNIX file.

System programmer response

If the file name is correct, use the FTP client's debug facility to determine why FTP cannot open the file.

Module

EZAFTPCK

EZA2739W **Translate file '*file*' invalid. SBDATACONN ignored.**

Explanation

A LOCSITE subcommand has been entered with an SBDATACONN parameter that specified a file name, 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.

Operator response

Ensure that the file name 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

Use the client's debug facility to help determine why the file cannot be used.

Module

EZAFTPCK

EZA2740E **SQL query not available. Can't load CAF routines.**

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.

Operator response

If FTP SQL support is required, ensure that the appropriate DSNLOAD library is included in the STEPLIB for the FTP client. (If FTP is started from the UNIX shell, the STEPLIB environment variable must be set.) Restart the FTP client session.

If FTP SQL support was not intended, issue a **locsite filetype=** command to change the file type from its current setting of SQL to SEQ.

System programmer response

Ensure that the appropriate DSNLOAD library is accessible.

Module

EZAFTPMQ

EZA2741W **Ctrlconn invalid syntax. Specify CTRLCONN=ascii_code_page_name or CTRLCONN=7BIT. CNTRLCONN ignored.**

Explanation

A LOCSITE 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.

Operator response

Reenter the LOCSITE command with corrected syntax.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2742W ***parameter=operand* ignored. Requested conversion is not supported.**

Explanation

A CTRLCONN parameter was entered on a LOCSITE subcommand, but there is no conversion available between the FTP client's code page and *operand*. The CTRLCONN parameter is ignored. The translate tables for the control connection are not changed.

System action

FTP continues.

Operator response

Reissue the LOCSITE command, specifying either "7bit" or the name of an ASCII code page supported by iconv. See the z/OS XL 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

Module

EZAFTPCK

EZA2743W ***keyword=value*** **ignored. Unable to set up requested conversion.**

Explanation

A CTRLCONN parameter was entered on a LOCSITE 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 client 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.

Operator response

No action is needed.

System programmer response

If the FTP client is running in a double-byte code set, the CTRLCONN parameter is not supported. The tables used for the control connection are set at client initialization from either a TCPXLBIN file, or from FTP's internal default single-byte tables.

If the client 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.

Module

EZAFTPCK

EZA2744W ***parameter*** **parameter already specified. "*parameter=value*" ignored.**

Explanation

parameter parameter was already specified on the LOCSITE command. This parameter can only be issued once per LOCSITE 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. The *value* specified for this occurrence of *parameter* is ignored.

Operator response

If you want to change the parameter value to the value specified by *parameter=value*, reissue the LOCSITE command with *parameter=value* as the only occurrence of the parameter on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2745I **Error trying to delete the new file.**

Explanation

FTP attempted to delete the new file that was created but the delete failed. One possible reason for a delete failure is that the data set has an unexpired retention period.

System action

No action is needed.

Operator response

The data set can be deleted with the data set utilities of the operating system.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2746E **A qualifier in *name* begins with an invalid character.**

Explanation

You entered a subcommand with *name* as a parameter. A data set qualifier specified in *name* begins with an invalid character.

System action

No action is needed.

Operator response

Reissue the subcommand with a valid data set name.

System programmer response

No action is needed.

Module

EZAFTPGU

EZA2747E **A qualifier in *name* is more than 8 characters.**

Explanation

A data set qualifier specified on the LCD or LMKDIR command is longer than 8 characters.

System action

No action is needed.

Operator response

Reissue the command with a valid data set name.

System programmer response

No action is needed.

Module

EZAFTPGU

EZA2748E **A qualifier in *name* contains an invalid character.**

Explanation

A data set qualifier specified on the current subcommand contains an invalid character.

System action

FTP rejects the subcommand. Processing continues.

Operator response

Reissue the subcommand with a valid data set name.

System programmer response

No action is needed.

Module

EZAFTPGU

EZA2749E **Cannot establish conversion between *codeset* and UCS-2.**

Explanation

While processing the FTP.DATA file, the value for the UCSHOSTCS parameter was not a valid code set name. An attempt to set up a conversion table between *codeset* and UCS-2 failed.

System action

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

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 code set for UCSHOSTCS. See the z/OS Communications Server: IP Configuration Reference for information on the parameters of the FTP.DATA file.

Module

EZAFTPCT, EZAFTPEP

EZA2756W **Unable to setup conversion between UCS-2 and *codeset*. UCSHOSTCS ignored.**

Explanation

A conversion between the codeset specified on the LOCSITE UCSHOSTCS command and UCS-2 could not be completed.

System action

The UCSHOSTCS parameter on the LOCSITE command is ignored. FTP continues.

Operator response

Reissue the LOCSITE command with a valid code set name.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2757I **Host code set for TYPE U transfer *code_set***

Explanation

This message is issued in response to the LOCSTAT subcommand when the FTP data transfer type is 'U' (UCS-2). It displays the name of the EBCDIC code set used for EBCDIC to UCS-2 conversion and UCS-2 to EBCDIC conversion.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2773E **TYPE U transfer is not supported in mode *mode***

Explanation

FTP has retrieved or sent an UCS-2 file (or data set) but the current mode is not set to Stream. Stream mode is required for UCS-2 data.

System action

The GET or PUT subcommand is rejected.

Operator response

Change the mode to stream using the STREAM command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2774E **TYPE U transfer is only supported in stream mode**

Explanation

To transfer a UCS-2 file, stream mode must be used. See EZA2773E for more information.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2775E **Record structure is not supported for TYPE U transfer**

Explanation

FTP has retrieved or sent an UCS-2 file (or data set) but the current structure is set to Record. File structure is required for UCS-2 data.

System action

The GET or PUT command is rejected.

Operator response

Change the structure to FILE using the STRUCT=F command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2776E **TYPE U transfer is not supported for filetype *type***

Explanation

FTP has retrieved or sent an UCS-2 file (or data set) but the current file type is not set to SEQ. Sequential file is required for UCS-2 data. JES or SQL files cannot be transferred as UCS-2 data.

System action

The GET or PUT command is rejected.

Operator response

If the file is not JES or SQL, change the file type to SEQ using the LOCSITE FILETYPE=SEQ command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2777I **TYPE U transfer is only supported for filetype SEQ**

Explanation

An UCS-2 transfer failed because the file type was not sequential. See EZA2776E for more information.

System action

FTP continues.

Operator response

None

System programmer response

None

Module

EZAFTPCG, EZAFTPCP

EZA2778E **Transfer of RDWs is not supported for TYPE U. Use 'LOCSITE NORDW'**

Explanation

FTP has sent an UCS-2 file (or data set) and RDWs is currently ON. RDWs must be off when sending UCS-2 data.

System action

The PUT command is rejected.

Operator response

Turn RDWs off using the LOCSITE NORDW command.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2779E **Transfer of trailingblanks is not supported for TYPE U. Use 'LOCSITE NOTRAILINGBLANKS'**

Explanation

FTP has sent an UCS-2 file (or data set) and TRAILINGBLANKS is currently ON. TRAILINGBLANKS must be off when sending UCS-2 data.

System action

The PUT command is rejected.

Operator response

Turn TRAILINGBLANKS off using the LOCSITE NOTRAILINGBLANKS command.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2780E **NOASATRANS is not supported for TYPE U. Use 'LOCSITE ASATRANS'**

Explanation

FTP has retrieved or sent an UCS-2 file (or data set) and ASATRANS is currently set to OFF. ASATRANS must be ON when storing UCS-2 data in EBCDIC format into an ASA data set.

System action

The GET or PUT command is rejected.

Operator response

Turn ASATRANS on using the LOCSITE ASATRANS command.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2781E **Transfer aborted: conversion failure**

Explanation

FTP has retrieved or sent an UCS-2 file (or data set), but was unable to convert it to/from EBCDIC.

System action

The GET or PUT command is rejected.

Operator response

Turn on debug and issue GET or PUT again. Give the output to the system programmer.

System programmer response

Check the debug messages to determine the errno and errnojr being returned from the iconv system call.

Module

EZAFTPCG, EZAFTPCP

EZA2782E **Transfer aborted: substitution occurred for TYPE U transfer**

Explanation

FTP has retrieved or sent an UCS-2 file (or data set) that contained a UCS-2 character that could not be converted to EBCDIC. The EBCDIC substitution character would need to be used to convert the UCS-2 character.

System action

The GET or PUT command is rejected.

Operator response

To allow the file to be stored in EBCDIC, substitution must be allowed. Issue LOCSITE UCSSUB to turn ON UCS-2 substitution.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2783E **Transfer aborted: uneven number of bytes received for TYPE U transfer**

Explanation

FTP has retrieved or sent an UCS-2 file (or data set) that contained an uneven number of bytes.

System action

The GET or PUT command is rejected.

Operator response

No action is needed.

System programmer response

Make sure the UCS-2 file is not corrupt. By definition UCS-2 is double byte, which should result in the file length being an even number of bytes.

Module

EZAFTPCG

EZA2784I **Conversion from Unicode to EBCDIC resulted in *number* substitutions.**

Explanation

FTP has received a Unicode file. Some characters were replaced with the EBCDIC substitution character.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2785E **WRAPRECORD is not supported for TYPE U. Use 'LOCSITE NOWRAPRECORD'**

Explanation

FTP has retrieved an UCS-2 file (or data set) and WRAPRECORD is currently ON. WRAPRECORD must be off when storing UCS-2 data in EBCDIC format.

System action

The GET command is rejected.

Operator response

Turn WRAPRECORD off using the LOCSITE NOWRAPRECORD command.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2786E **TYPE U transfer requires variable format data set**

Explanation

FTP has retrieved an UCS-2 file (or data set) but the target data set is not variable format.

System action

The GET command is rejected.

Operator response

Change the target data set to a variable format data set.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2787E **Transfer aborted: truncation occurred for TYPE U transfer**

Explanation

FTP has retrieved an UCS-2 file (or data set) and converted it to EBCDIC. Some of the records were truncated. Also UCSTRUNC is currently set to NO.

System action

The GET command is rejected.

Operator response

Either set UCSTRUNC to YES using the LOCSITE UCSTRUNC command, or increase the logical record length of the retrieved file.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2788I **Command ! may only be issued when the client is running in the z/OS UNIX shell.**

Explanation

This message is issued when the ! subcommand was issued while not under the z/OS UNIX System Services shell. The ! subcommand is not supported in a batch, EXEC, TSO, or Callable API environment.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2789E **DSORG=DA requires structure: FILE, mode: STREAM, and transfer type: IMAGE.**

Explanation

A PUT command was issued for a DSORG=DA data set, but one or more of the following conditions are not met: FILE structure, STREAM mode, and IMAGE transfer type.

System action

FTP continues.

Operator response

If you intend to transfer a DSORG=DA file, use the LOCSTAT command to display the current status for the FTP client. Then use the STRUCT F, MODE S, and/or TYPE I commands as required.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2790E **LRECL X valid only for RECFM of U or variable length spanned.**

Explanation

The LOCSITE command was issued with LRECL=X, but the record format is not U or variable length spanned.

System action

FTP continues.

Operator response

If you intend to set LRECL=X, use the LOCSITE command to set the record format to U or variable length spanned.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2791I **LRECL being reset to *lrecl*.**

Explanation

The LOCSITE command was issued with LRECL=X, but the record format is not U or variable length spanned. LRECL is being reset to the previous value or 256 if the previous value was invalid.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2792I **LRECL being reset to site default of 'not used'.**

Explanation

The LOCSITE command was issued with LRECL=X, but the record format is not U or variable length spanned. LRECL is being reset to the default value of 'not used'.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2793I **UCS Substitution: *ucssub*, UCS Truncation: *ucstrunc***

Explanation

This message is issued in response to the LOCSTAT subcommand if the transfer type is UCS-2. The ucssub value indicates whether substitution is allowed when converting between UCS-2 and EBCDIC. The ucstrunc value indicates whether truncation is allowed when converting between UCS-2 and EBCDIC.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2794I	Both CCTRANS and CTRLCONN were specified. Using CTRLCONN. CCTRANS will be ignored.

Explanation

The FTP client has encountered both CCTRANS and CTRLCONN keywords in FTP.DATA. The CCTRANS keyword is supported only to aid migration; the CTRLCONN keyword is the preferred keyword. When both keywords are present in FTP.DATA, the CTRLCONN value will be used for the client configuration and the CCTRANS value will be ignored.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

To avoid this message, remove the CCTRANS statement from the FTP.DATA file.

Module

EZAFTPEP

EZA2795I	Both SBTRANS and SBDATACONN were specified. Using SBDATACONN. SBTRANS will be ignored.

Explanation

The FTP client has encountered both SBTRANS and SBDATACONN keywords in FTP.DATA. The SBTRANS keyword is supported only to aid migration; the SBDATACONN keyword is the preferred keyword. When both keywords are present in FTP.DATA, the SBDATACONN value will be used for the client configuration and the SBTRANS value will be ignored.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

To avoid this message, remove the SBTRANS statement from the FTP.DATA file.

Module

EZAFTPEP

EZA2798I **Byte Order: *byte_order***

Explanation

This message is issued in response to the LOCSTAT subcommand if the transfer type is UCS-2. The byte_order indicates whether the UCS-2 files sent by the client are encoded using big-endian or little-endian.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2799W **The data set is allocated to another job and is unavailable.**

Explanation

The client was unable to allocate the data set because it is already allocated to another job.

System action

No action is needed.

Operator response

Try again later.

System programmer response

No action is needed.

Module

EZAFTPCP, EZAFTPCG

EZA2801I **Data not wrapped into next record.**

Explanation

Data that does not fit within the target record will be truncated.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2802I **Data was truncated.**

Explanation

A data transfer has completed, but FTP was not able to store all the data because the logical record size for the receiving data set was not large enough. Some data has been lost.

System action

FTP continues.

Operator response

If truncation is not acceptable, redo the data transfer to an existing data set that has a larger record size, or you can use the LOCSITE LRECL parameter to specify a particular record size that is larger and redo the data transfer to a new data set.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2803I **Lines were wrapped.**

Explanation

A data transfer has completed. One or more lines of data were longer than the record size of the receiving data set; those lines were continued in the next record.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2804W **Data set is out of space.**

Explanation

A data transfer has failed because there is not enough space in the data set.

System action

FTP continues.

Operator response

See message EZA2805W for the system completion code and reason code and take actions appropriate for the failure.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2805W **System completion code and reason: *cc-rc***

Explanation

A file I/O operation failed with the system completion code *cc* and the reason code *rc*.

cc-rc is the hexadecimal system completion code and the return code. See the z/OS MVS System Codes for the descriptions of the system completion codes. Return codes are described as part of the system completion codes.

System action

FTP continues.

Operator response

See the z/OS MVS System Codes publication for suggested actions for the specified failure code.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

EZA2806I **DBCS conversion to EBCDIC encountered invalid input (*number* errors)**

Explanation

The FTP client encountered input errors during conversion of type B data.

System action

FTP continues.

Operator response

For more specific information about the errors, issue the DEBUG subcommand and then reissue the GET subcommand. Check the source data and correct any errors. If you are creating a pure DBCS local file, ensure that the source data does not contain any single-byte data. Resubmit the data transfer.

System programmer response

No action is needed.

Module

EZAFTPCG

EZA2807I **Executing under single stack configuration. Specified TCPIP name *tcpipname* ignored.**

Explanation

The FTP client was started with the **-p** or (TCP start parameter that specified TCPIP stack *tcpipname* was to be used. The client is running in an INET (single-stack) configuration and this start parameter applies only to a CINET (multi-stack) configuration.

System action

The parameter is ignored. FTP client continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCY

EZA2808W **File has unsupported DSORG.**

Explanation

Requested file has a DSORG that is not supported by FTP.

System action

The file transfer request is terminated. FTP continues.

Operator response

Check DSORG for the file.

System programmer response

No action is needed.

Module

EZAFTPCP

EZA2809I **CCONNTIME is *cconntime***

Explanation

CCONNTIME is the amount of time to wait after attempting to close a control connection before terminating the control connection and reporting an error.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2810I **DATACTTIME is *datacttime***

Explanation

DATACTTIME is the amount of time to wait after attempting to send or receive data before terminating the data connection and reporting an error to the user.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2811I **DCONNTIME is *dconntime***

Explanation

DCONNTIME is the amount of time to wait after attempting to close a data transfer connection before terminating the data connection and reporting an error.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2812I **INACTTIME is *inacttime***

Explanation

INACTTIME is the amount of time to wait for an expected response from the server, on either the control or the data connection, before closing the session.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2813I **MYOPENTIME is *myopentime***

Explanation

MYOPENTIME is the amount of time to wait for a session to open before terminating the attempt and reporting an error.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2814I **UCOUNT is *ucount***

Explanation

UCOUNT is the unit count for new allocations.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2815I **VCOUNT is *vcount***

Explanation

VCOUNT is the volume count for new allocations.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

EZAFTPCL

Module

No action is needed.

EZA2816I **No automatic mount of tape volumes.**

Explanation

NOAUTOMOUNT was specified. NOAUTOMOUNT is the default value.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2817I **Automatic mount of tape volumes.**

Explanation

AUTOMOUNT was specified.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2818I **Data connections for the client are firewall friendly.**

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates that when a data connection is set up, the client will be the active end of the data connection and therefore firewall friendly.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2819I **Data connections for the client are not firewall friendly.**

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates that when a data connection is set up, the client will be the passive end of the data connection and therefore not firewall friendly.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

EZA2822W **NOFWFriendly already specified. FWFriendly ignored.**

Explanation

NOFWFriendly was already specified on this LOCSITE command, FWFriendly is ignored.

System action

FTP continues.

Operator response

Specify one option, FWFriendly or NOFWFriendly on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2823W **FWFriendly already specified. NOFWFriendly ignored.**

Explanation

FWFriendly was already specified on this LOCSITE command, NOFWFriendly is ignored.

System action

FTP continues.

Operator response

Specify one option, FWFriendly or NOFWFriendly on the LOCSITE command.

System programmer response

No action is needed.

Module

EZAFTPCK

EZA2824W **Open of *filename* failed due to incorrect BLKSIZE or LRECL. Use SITE LRECL or BLKSIZE**

Explanation

An incorrect block size or logical record length was specified for the file.

filename is the name of the file that could not be opened.

System action

The requested operation is ended. FTP continues.

Operator response

Override the incorrect BLKSIZE/LRECL by issuing a SITE command with the correct settings.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndfile

EZA2825E **File contains records that are too long to be processed**

Explanation

The FTP client cannot process the received data because a record delimiter was not found in the data and the FTP client is not configured to wrap records. The FTP client expects to find a record delimiter, such as a Carriage Return and Line Feed (CRLF) in ASCII mode or a Newline(NL) in EBCDIC mode, to denote the end of a record.

System action

The file is not transferred.

Operator response

Ensure that the ASCII or EBCDIC file being received has proper delimiters in the data or configure the FTP client to wrap records using the wraprecord parameter. See z/OS Communications Server: IP User's Guide and Commands for information about configuring wraprecord.

System programmer response

No action is needed.

Module

EZAFTPCG

Procedure name

mvs_rcvFile()

EZA2826E **Volume=*volume_serial_list* contains a tape volume.**

Explanation

The LMKDIR subcommand was issued to create a new PDS. The LOCSITE VOLUME parameter was in effect, specifying that all new data sets will be allocated on volumes *volume_serial_list*. One of the volumes in *volume_serial_list* was determined to be a tape, and partitioned data sets cannot be allocated to tape.

System action

The LMKDIR subcommand is rejected.

Operator response

Issue the LOCSITE VOLUME command to change the *volume serial list* for new data sets to DASD volumes, then reissue the LMKDIR subcommand. See the z/OS Communications Server: IP User's Guide and Commands for more information on the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCD

Procedure name

lmkdir()

EZA2827I **Error mounting one of these volumes: *volume_serial_list*.**

Explanation

The LMKDIR subcommand was issued to create a new PDS. The LOCSITE VOLUME parameter was in effect, specifying that all new data sets will be allocated on volumes *volume_serial_list*. FTP was not able to mount one of the volumes in *volume_serial_list*.

System action

The LMKDIR subcommand is rejected.

Operator response

Ask the operator to mount all volumes required. Reissue the LMKDIR command. If this does not correct the problem, report the error to the system programmer.

System programmer response

Check the MVS console for messages related to the volumes in *volume_serial_list*. Correct the errors, and reissue LMKDIR.

Module

EZAFTPCD

Procedure name

lmkdir()

EZA2828I **Volume=*volume_serial_list* contains a volume that is not mounted.**

Explanation

The LMKDIR subcommand was issued to create a new PDS. The LOCSITE VOLUME parameter was in effect, specifying that all new data sets will be allocated on volumes *volume_serial_list*. One or more or the volumes in *volume_serial_list* was not mounted, and NOAUTOMOUNT was in effect. NOAUTOMOUNT prevents automatic mounting of volumes that are not mounted.

System action

The LMKDIR subcommand is rejected.

Operator response

Ask the operator to mount all required volumes, or issue the LOCSITE AUTOMOUNT subcommand to permit automatic mounting of volumes that are not mounted. Reissue the LMKDIR subcommand. See the z/OS Communications Server: IP User's Guide and Commands for more information on the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCD

Procedure name

lmkdir()

EZA2829W **UCOUNT value *(value)* must be P, or an integer from *minimum* through *maximum*.**

Explanation

The LOCSITE subcommand was issued with the UCOUNT parameter. The value, *value*, specified for the UCOUNT parameter was not an integer from *minimum* through *maximum*, nor was it the letter P.

System action

The UCOUNT parameter is ignored.

Operator response

Reissue the command with a valid value.

System programmer response

No action is needed.

Module

EZAFTPCK

Procedure name

locsite()

EZA2830I **Anonymous FTP log in from *ipaddr* at *currenttime* - email address: *emailaddress***

Explanation

An anonymous user logged in and it is written to syslog.

ipadd is the IP address of the client logging in to FTP.

currenttime is the time the login occurred.

emailaddress is what the user entered as his or her email address.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPRA

EZA2831E **EMAILADDRCHECK value must be NO, WARNING or FAIL**

Explanation

While processing the FTP.DATA file, the server encountered the EMAILADDRCHECK parameter with a parameter value that was not NO, WARNING or FAIL. The only valid values for the EMAILADDRCHECK parameter are NO, WARNING or FAIL.

System action

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

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. See the z/OS Communications Server: IP Configuration Reference for information on the parameters of the FTP.DATA file.

Module

EZAFTPEP

EZA2832I **Load module transfer requires FILETYPE = SEQ**

Explanation

The user has requested a file transfer that might involve load modules and the client's local file type is not SEQ.

System action

Load module transfer processing ends and base processing continues.

Operator response

If you want load module transfer processing to be performed, issue a LOCSITE FILETYPE=SEQ subcommand.

System programmer response

No action is needed.

Module

EZAFTPCC

EZA2833I **Load module transfer does not support load module rename**

Explanation

The user has requested a file transfer that might involve load modules and has requested that the files be renamed. Load module transfer processing does not support renaming of load modules.

System action

Load module transfer processing ends and base processing continues. Any MVS load modules transferred will not be executable on the target system.

Operator response

If you want load module transfer processing to be performed, issue a transfer subcommand that does not include a rename of the files.

System programmer response

No action is needed.

Module

EZAFTPCC

EZA2834I **Load module transfer supports member names only**

Explanation

The user has requested a file transfer that might involve load modules but has specified either an z/OS UNIX file name or a fully qualified MVS data set name. Load module transfer only supports transfer of members of the PDS or PDSE which is the current working directory.

System action

Load module transfer processing ends and base processing continues. Any MVS load modules transferred will not be executable on the target system.

Operator response

If you want load module transfer processing to be performed, reissue the transfer subcommand without using quotes or slashes.

System programmer response

No action is needed.

Module

EZAFTPCC

EZA2835I **Load module transfer requires local directory to be PDS or PDSE**

Explanation

The user has requested a file transfer that might involve load modules but the current working directory is not a PDS or a PDSE. Load module transfer requires the current working directory to be a PDS or a PDSE.

System action

Load module transfer processing ends and base processing continues. Any MVS load modules transferred will not be executable on the target system.

Operator response

If you wanted load module transfer processing to be performed, change the local directory to a load library (a PDS or a PDSE with RECFM=U).

System programmer response

No action is needed.

Module

EZAFTPCC

EZA2836I **No members of the data set match the selection pathnames**

Explanation

The user has requested a PUT or MPUT that might involve load modules but the PDS or PDSE current working directory does not contain any members that match the local file name provided.

System action

FTP waits for the next subcommand.

Operator response

Check the file name specified on the PUT or MPUT subcommand.

System programmer response

No action is needed.

Module

EZAFTPCC

EZA2837I **Could not build a file list**

Explanation

While performing load module transfer, the FTP client was unable to build a list of file names to transfer.

System action

Load module transfer processing ends and base processing continues. Any MVS load modules transferred will not be executable on the target system.

Operator response

No action is needed.

System programmer response

Verify adequate storage available for memory files.

Module

EZAFTPCC

EZA2838I **Load module transfer syntax error**

Explanation

An unexpected or incorrectly formatted load module transfer response was sent from the FTP server to the FTP client. This is a violation of the load module transfer protocol.

System action

Load module transfer processing is ended and the file transfer is completed using base processing. Any MVS load modules transferred will not be executable on the target system.

Operator response

Contact the system programmer.

System programmer response

Ensure that the server and client are both at a supported level of Communications Server. If not, contact the vendor of the non-IBM FTP program for service.

Module

EZAFTPCC

EZA2839I **Reload of the load library failed**

Explanation

While processing a GET or MGET command for one or more load modules, the client failed to reload the unloaded load module from the temporary data set into the load library.

System action

The transfer is ended, and no files will be transferred. FTP waits for the next subcommand.

Operator response

Contact the system programmer.

System programmer response

Verify that sufficient free space exists in the load library to receive the load modules. Look for console messages from the file system or the IEBCOPY system utility. If none were provided, get an FTP client trace of the operation.

Module

EZAFTPCC

EZA2840I **Unload of the load library failed**

Explanation

While processing a PUT or MPUT command for one or more load modules, the client failed to unload the load module from the load library into a temporary data set.

System action

The transfer is ended, and no files will be transferred. FTP waits for the next subcommand.

Operator response

Contact the system programmer.

System programmer response

Verify sufficient free space exists in the file system for a temporary file to hold the unloaded load modules. Look for console messages from the file system or the IEBCOPY utility. If none were provided, get an FTP client trace of the operation.

Module

EZAFTPCC

EZA2841I **Local directory might be a load library**

Explanation

As a result of the LCD subcommand, the FTP client local directory is a PDS or a PDSE with RECFM=U. This might be a load library, and load module transfer processing might be attempted for this directory.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCD

EZA2842I **System service invoked for load module transfer failed: IEBCOPY rc=*rc***

Explanation

While performing a load module transfer, the client invoked the IEBCOPY system utility to unload load modules into a temporary data set (for PUT or MPUT) or to reload unloaded load modules from temporary data set into the target load library (for GET or MGET). IEBCOPY failed.

rc is the decimal return code provided by IEBCOPY. IEBCOPY return codes are documented in z/OS DFSMSdfp Utilities.

System action

Load module transfer ends. No files will be transferred. The FTP client waits for the next subcommand.

Operator response

Contact the system programmer

System programmer response

Use the z/OS DFSMSdfp Utilities manual to determine the corrective action for the IEBCOPY return code.

Module

EZAFTPMD

EZA2843E **Incorrect format data set name *dsn***

Explanation

The FTP server was parsing keywords in the FTP.DATA file. The data set name specified, *dsn*, was not a valid MVS data set name. The keyword is ignored. The FTP server continues with the next keyword.

System action

FTP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the data set name in the FTP.DATA file. See the z/OS Communications Server: IP Configuration Reference for information on the parameters of the FTP.DATA file.

Module

EZAFTPEP

EZA2844I **ISPFSTAT is set to *ispfstat***

Explanation

ISPFSTAT can be set to either TRUE or FALSE. When ISPFSTAT is set to TRUE, ISPF Statistics of a PDS member that has been transferred to the FTP client will be updated or created. The default setting is FALSE.

ispfstat is either TRUE or FALSE.

System action

FTP Continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2845I **Statistics have been updated for the PDS member that was transferred**

Explanation

Statistics had to be updated for the PDS member that was transferred because the PDS member that it replaced had statistics. ISPFSTAT setting of FALSE was ignored.

System action

FTP continues

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPMR

Procedure name

pds_stats_update()

EZA2846E **Lrecl X transfer not supported with *condition***

Explanation

A put was issued for a data set with Lrecl X and an incorrect condition was detected.

condition can be one of the following:

MODE C
Compressed mode

TYPE U 2
Data type U 2

FILETYPE=SQL
SQL

NOASATRANS
NOASAtrans cannot be specified with Lrecl X w/ASA

STRUCT R
Record structure

System action

The request is rejected. FTP continues.

Operator response

Send the Lrecl X data set using the correct FILETYPE, MODE, STRUCT, TYPE, or ASATRANS setting.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndFile

EZA2850E **Lrecl X transfer not supported with Record I/O**

Explanation

A put was issued for a data set with lrecl X and Record I/O has been requested. The condition is unexpected.

System action

The request is rejected. FTP continues.

Operator response

Contact the system programmer.

System programmer response

Collect an FTP Client trace that includes the failure and call the IBM Support Center.

Module

EZAFTPCP

Procedure name

sndFile

EZA2852I **Parameters not supported: *value_1 ... value_n***

Explanation

This is a list of the values entered on a DEBUG or a DUMP subcommand that are not supported.

System action

The unsupported values are ignored. FTP continues.

Operator response

See the z/OS Communications Server: IP User's Guide and Commands for information about FTP Subcommands and the supported values for the DEBUG and DUMP subcommands.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

debug

EZA2858I **Message IDs are not displayed when running in *system* UNIX**

Explanation

This message is generated after the VERBOSE subcommand to the FTP client. It indicates that VERBOSE mode is toggled off and client messages will be displayed without their message ID numbers. The VERBOSE setting is applicable only when the FTP client is running in a UNIX environment.

system is the name of the operating system (for example, z/OS).

System action

FTP continues.

Operator response

Reissue the VERBOSE subcommand to toggle the setting on, if necessary.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

verbose()

EZA2859I **Message IDs are displayed when running in *system* UNIX**

Explanation

This message is generated after the VERBOSE subcommand to the FTP client. It indicates that VERBOSE mode is toggled on and client messages will be displayed with their message ID numbers. The VERBOSE setting is applicable only when the FTP client is running in a UNIX environment.

system is the name of the operating system (for example, z/OS).

System action

FTP continues.

Operator response

Reissue the VERBOSE subcommand to toggle the setting off, if necessary.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

verbose()

EZA2860I **Secure Hostname is: *level***

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates whether host name verification in the server's certificate is required or optional.

In the message text:

level
The setting for the SECURE_HOSTNAME statement in the FTP.DATA file. The possible levels for the secure host name are REQUIRED or OPTIONAL. OPTIONAL is the default.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2861I **Using *socks_configuration_file* for SOCKS server configuration**

Explanation

The FTP client identified a file or data set containing SOCKS server configuration information.

socks_configuration_file is the z/OS UNIX file or MVS data set containing SOCKS configuration information

System action

The FTP client continues processing. The SOCKS configuration will be referenced for any attempt to connect the client to an FTP server. The SOCKS configuration might cause the FTP client to connect to FTP servers through a SOCKS server, instead of connecting directly to the FTP server.

Operator response

If *socks_configuration_file* is the correct file to use for the SOCKS server configuration, no action is needed. Otherwise, contact the system programmer.

System programmer response

If *socks_configuration_file* is the correct file to use for the SOCKS server configuration, no action is needed. If you want to use a different file for the SOCKS configuration, see the z/OS Communications Server: IP Configuration Reference for information about configuring the FTP client for SOCKS.

Module

ezaftpcl

Procedure name

locstat()

EZA2862I **Errors parsing SOCKS configuration file *file* on line *line_number* - use of SOCKS has been disabled.**

Explanation

The FTP client located the specified SOCKS configuration file, but encountered syntax errors when parsing the information.

file is the name of the z/OS UNIX file or MVS data set containing the SOCKS configuration file.

line_number is the line of file where the error was encountered.

System action

The FTP client continues, but disables use of any SOCKS server for this session.

Operator response

Contact the system programmer.

System programmer response

Correct the error in the SOCKS configuration file or data set. See the z/OS Communications Server: IP Configuration Reference manual for information about configuring the FTP client for SOCKS.

Module

ezaftpcy

Procedure name

ftpSocks()

EZA2863I **Error *error* opening SOCKS configuration file *socks_configuration_file* - use of SOCKS has been disabled**

Explanation

The FTP client encountered an error while attempting to open the SOCKS configuration file.

error is the C run-time library error message returned for the failing fopen() call. For more information, see z/OS Language Environment Runtime Messages.

socks_configuration_file is the name of the file that failed to open.

System action

The FTP client continues. It will not be configured to use a SOCKS server.

Operator response

Specify the correct *socks_configuration_file* or contact the system programmer.

System programmer response

Specify the correct *socks_configuration_file* or correct the error indicated by *error*.

Module

EZAFTPCY

Procedure name

findSocks()

EZA2864I **Error *error* reading SOCKS configuration file *socks_configuration_file* - use of SOCKS has been disabled**

Explanation

The FTP client encountered an error while attempting to read the SOCKS configuration file.

error is the C run-time library error message returned. For more information, see z/OS Language Environment Runtime Messages.

socks_configuration_file is the name of the file that had the read error.

System action

The FTP client continues. It will not be configured to use a SOCKS server.

Operator response

Specify the correct *socks_configuration_file* or contact the system programmer.

System programmer response

Specify the correct *socks_configuration_file* or correct the error indicated by *error*.

Module

EZAFTPCY

Procedure name

ftpSocks()

EZA2865I **Tape write is allowed to use BSAM I/O.**

Explanation

When processing ASCII data in stream mode for GET and writing to tape, the FTP Client is allowed to use the BSAM I/O routine instead of the LE Run-Time Library. The WRTAPEFastio option has been set in the FTP.DATA data set or in a LOCSITE subcommand. The option is honored unless a request that contradicts it has been made (such as ASATRANS for a data set with ASA control characters).

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2866I **Tape write is not allowed to use BSAM I/O.**

Explanation

When processing ASCII data in stream mode for GET and writing to tape, the FTP Client must use the LE Run-Time Library fwrite() function. This is the default setting and is changed by specifying WRTAPEFastio TRUE in the FTP.DATA data set or by specifying WRTAPEFastio on a LOCSITE subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

EZA2870I **TLS security mechanism negotiation failed - data connection closed**

Explanation

The FTP client and server failed during the negotiation step (also known as the handshake) to protect a data connection with the TLS mechanism.

System action

No action is needed.

Operator response

Activate the FTP client and server traces as follows:

```
SITE DEBUG=(NONE,ACC,SOC(3))
DEBUG NONE ACC SOC(3)
```

Try the subcommand again and contact the system programmer with the error message.

If you are not allowed to issue the SITE command to change the server trace options, ask the system programmer to set the options, close the session to the server, enter FTP again and try the failing subcommand.

System programmer response

Use the FTP client and server traces to determine the cause of the error and correct the problem.

Module

EZAFTPCA, EZAFTPCC, EZAFTPCG, EZAFTPCP

Procedure name

getList, rcvXLMT, sndXLMT, rcvFile, rcvNullDir, sndFile

EZA2889I **Authentication mechanism: *mechanism***

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates that the FTP session is protected by the authentication mechanism that is displayed.

The supported mechanisms are GSSAPI or TLS.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2890I **Control connection protection: *level***

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates that the control connection has the security level that is displayed.

The possible security levels for the GSSAPI security mechanism are Clear, Safe, or Private.

The security level for the TLS security mechanism is always Private.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2891I **Data connection protection: *level***

Explanation

This message is issued in response to the LOCSTAT subcommand. It indicates that the data connection has the security level that is displayed.

The possible security levels for the GSSAPI security mechanism are Clear, Safe, or Private.

The possible security levels for the TLS and TLS-P security mechanisms are Clear and Private.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2892I **Secure port *port_number* does not support the -a or -r start parameter**

Explanation

You issued the **`ftp`** command with the port_number parameter and the -a or -r parameter. The port number you specified was the secure port. The -a and -r start parameters are not allowed when you specify the port indicated by the *port_number* value.

In the message text:

port_number
The port number on the remote host to which you specified that the FTP client should be connected.

System action

The FTP client ends.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Issue the **ftp** command without the -a or -r start parameter to log into the server using the secure port. Or, log into the server using the -a or -r start parameter with another port number.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: FTP

Module

Not applicable.

Routing code

10

Descriptor code

12

Example

Your client FTP.DATA contains this statement:

```
TLSPORT 990
```

This statement sets the secure port to 990. You attempt to connect to the FTP server at port 990 on the host specified *hostname* value. You specify the -a parameter also:

```
/rpcbind ftp -v  -a tls hostname 990
EZA2892I Secure port 990 does not allow the -a or -r start parameter
```

Message EZA2892I is issued because you specified the -a parameter while trying to connect to the secure port.

EZA2894I **Security mechanism *mechanism* is not valid and is ignored**

Explanation

You specified the mechanism on the **-a** or the **-r** parameter on the FTP command. The mechanism *mechanism* is not one of the supported mechanisms.

System action

FTP continues.

Operator response

If a security mechanism is required, stop FTP and start again with one of the supported security mechanisms. The supported security mechanisms are GSSAPI or TLS.

System programmer response

No action is needed.

Module

EZAFTPCY

Procedure name

processStartParms

EZA2895I **Authentication negotiation succeeded**

Explanation

A security mechanism was requested to protect the session. The client and server have successfully negotiated to protect the session.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2896I **Unable to set control connection protection to required level *level***

Explanation

The client was unable to set the control connection protection to the required level as specified on the SECURE_CTRLCONN statement in the client's FTP.DATA file. The server does not support the protection level for the control connection.

System action

FTP continues. The connection to the FTP server ends.

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 set the SECURE_CTRLCONN statement to a level supported by the server. See the z/OS Communications Server: IP Configuration Guide for information about configuring FTP.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2897I **Authentication negotiation failed**

Explanation

The client was unable to negotiate authentication with the server.

System action

FTP continues.

Operator response

Activate the FTP client and server traces as follows:

- SITE DEBUG=(NONE,SEC,SOC(3))
- DEBUG NONE SEC SOC(3)

Try the connection again and contact the system programmer with the error message.

If you are not allowed to issue the SITE command to change the server trace options, ask the system programmer to set the options, close the session to the server, enter FTP again and try the failing subcommand.

System programmer response

Use the MODIFY command to set the tracing options for the server.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2898I **Unable to successfully negotiate required authentication**

Explanation

The client was unable to successfully negotiate the required authentication with the server.

System action

The connection with the server ends. FTP continues.

Operator response

Activate the FTP server and client traces as follows:

- SITE DEBUG=(NONE,ACC,SOC(3))
- DEBUG NONE ACC SOC(3)

Try the connection again and contact the system programmer with the error message.

If you are not allowed to issue the SITE command to change the server trace options, ask the system programmer to set the options. Then close the session to the server, restart FTP, and try the failing subcommand.

If the server is not the z/OS FTP server, get servers traces as defined by the operating system of the FTP server.

System programmer response

Contact your IBM Support Representative and provide the client and server traces for diagnosis.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2899I **Request to import name *name* failed**

Explanation

The call to the gss_import_name() function failed to import *name*. This message is preceded by the messages EZA2912I and EZA2913I. See those messages for further diagnosis.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

See z/OS Integrated Security Services Network Authentication Service Programming for an explanation of the gss_import_name() function.

Module

EZAFTPFC

Procedure name

ftpAuth

EZA2900I **Request to initialize the security context failed**

Explanation

The call to the gss_init_sec_context() function failed. This message is preceded by the messages EZA2912I and EZA2913I. See those messages for further diagnosis.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

See z/OS Integrated Security Services Network Authentication Service Programming for an explanation of the gss_init_sec_context() function.

Module

EZAFTPFC

Procedure name

ftpAuth

EZA2903I **The subcommand is not valid when there is no active security mechanism**

Explanation

You entered a subcommand that is not valid when no security mechanism is active.

System action

FTP continues.

Operator response

See the z/OS Communications Server: IP User's Guide and Commands for information about the valid security mechanisms for the subcommand.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

setclevel, setdlevel

EZA2904I **Cannot set protection level to *level***

Explanation

The protection was not set to the requested level. This might happen because the FTP.DATA file SECURE_CTRLCONN or SECURE_DATACONN statements do not support the requested level or because the server denied the request.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

setclevel, setdlevel

EZA2905I **Control connection protection is *level***

Explanation

This message displays the current level of protection for the control connection.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

setclevel

EZA2906I **Data connection protection is *level***

Explanation

This message displays the current level of protection for the data connection.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

setdlevel

EZA2907I **Attempt to encrypt data failed**

Explanation

The call to the gss_wrap() function failed. This message is preceded by the messages EZA2912I and EZA2913I. See those messages for further diagnosis.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

See z/OS Integrated Security Services Network Authentication Service Programming for an explanation of the gss_wrap() function.

Module

EZAFTPFC

Procedure name

gss_secure_command

EZA2909I **Cannot base 64 decode reply: *error***

Explanation

A reply from the server was unable to be Base 64 decoded.

error will be one of the following:

1

Bad character in encoding: An invalid character was received from the server as part of a base 64 encoded string. The valid characters in a base 64 encoded string are the letters A - Z, the letters a - z, the numbers 0 - 9, the plus sign (+), and the forward slash (/).

2

Encoding not properly padded: An invalid base 64 encoded string was received from the server. The string was not padded with equal signs to extend the length of the string to a multiple of four characters.

3

Decoded number of bits not a multiple of 8: An invalid base 64 encoded string was received from the server. The string was unable to be decoded because it did not decode into a multiple of eight bits.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

gss_secure_reply

EZA2910I **Attempt to decrypt data failed**

Explanation

The call to the gss_unwrap() function failed. This message is preceded by messages EZA2912I and EZA2913I. See those messages for further diagnosis.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

See z/OS Integrated Security Services Network Authentication Service Programming for an explanation of the gss_unwrap() function.

Module

EZAFTPFC

Procedure name

gss_secure_reply

EZA2911I **Received unprotected reply - Current protection level: *level***

Explanation

An unprotected reply was received from the server, but the current level of protection on the control connection required protection.

level is the current level of protection on the control connection and will be either Safe or Private.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

gss_secure_reply

EZA2912I **GSSAPI error major status code: *value* - *text***

Explanation

This message displays the status code returned from a GSSAPI function call. This message is followed by message EZZ2913 and a message indicating the failing GSSAPI function call.

value is the hexadecimal value of the major status code.

text is the text describing the status code obtained by calling the gss_display_status() function.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

See z/OS Integrated Security Services Network Authentication Service Programming for an explanation of the major status code.

Module

EZAFTPFC

Procedure name

gss_user_error

EZA2913I **GSSAPI error minor status code: *value* - *text***

Explanation

This message displays the minor status code returned from a GSSAPI function call. This message is preceded by message EZZ2912 and is followed by a message indicating the failing GSSAPI function call.

value is the hexadecimal value of the minor status code.

text is the text describing the status code obtained by calling the gss_display_status() function.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

See z/OS Integrated Security Services Network Authentication Service Programming for an explanation of the minor status code.

Module

EZAFTPFC

Procedure name

gss_user_error

EZA2915I **Length *length* of received buffer is greater than PBSZ *size***

Explanation

An encrypted message was received whose length was greater than the negotiated protection buffer size as specified on the SECURE_PBSZ statement in FTP.DATA.

length is the length of the received encrypted message.

size is the maximum length of an encrypted message as negotiated with the server.

System action

FTP continues. The message is discarded.

Operator response

Increase the SECURE_PBSZ value in the FTP.DATA file.

System programmer response

No action is needed.

Module

EZAFTPFC

Procedure name

gss_receive_data

EZA2916I **local site variable *statement_name* is set to *value***

Explanation

This message is displayed as part of the **LOCSTat** subcommand output.

statement_name is the name of the statement coded in the client FTP.DATA file.

value is the current setting of *statement_name*. It is the default value assigned by FTP, or what you specified in the FTP.DATA file, or the most recent value you assigned with a **LOCSIte** subcommand. You can use the LOCSIte subcommand to change *value*. See the z/OS Communications Server: IP User's Guide and Commands and the z/OS Communications Server: IP Configuration Reference for information about the *statement_name* statement in the client FTP.DATA file.

System action

Processing continues.

Operator response

If you want to change the setting of *statement_name* temporarily, use the **LOCSIte** subcommand. See the z/OS Communications Server: IP User's Guide and Commands for information about the **LOCSIte** subcommand. If you want to make a permanent change to *statement_name*, see the z/OS Communications Server: IP Configuration Reference for information about coding the *statement_name* statement.

System programmer response

No action is needed.

Module

ezaftpcl

Procedure name

locstat()

EZA2917I **PROXY OPEN is not supported with security mechanisms**

Explanation

The FTP session is protected by the GSSAPI security mechanism or the TLS security mechanism. FTP does not allow a second session to be opened using PROXY OPEN while the primary session is protected.

System action

FTP continues.

Operator response

If proxy transfers are needed, close the primary connection and open a connection that does not have a security mechanism protecting the session. Then the PROXY OPEN can be issued.

System programmer response

No action is needed.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2918I **Filetype SQL requires type A (ASCII) or E (EBCDIC)**

Explanation

The user issued a PUT or APPEND subcommand. The file type that is specified by the LOCSITE subcommand or in the FTP.DATA file is SQL. An SQL transfer from the client requires that the data type of the connection is ASCII or EBCDIC.

System action

The PUT or APPEND subcommand is not performed.

Operator response

Change the data type by issuing the ASCII or EBCDIC subcommand and reissue the put or append subcommand.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

sndFile

EZA2919I **Session starts with protection on the data connection**

Explanation

The data connection is protected for secure data transfers.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2920I **SECURE_DATACONN NEVER configuration statement is ignored**

Explanation

The mechanism TLS-P started with an implied protection level of PRIVATE which is inconsistent with the statement SECURE_DATACONN NEVER that is specified in the FTP.DATA file. The statement is ignored and a statement of SECURE_DATACONN CLEAR is assumed for this connection.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCZ

Procedure name

ftpOpen

EZA2922I **Active client dumpIDs - *id_1 ... id_n***

Explanation

This is a display of the active FTP extended trace IDs (dumpIDs).

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

dump

EZA2923I **Active client traces - *name_1 ... name_n***

Explanation

This message displays the active FTP client general traces. This message is issued when a debug subcommand is used to change or display the values for the FTP general trace.

name_x can be one of the following:

FLO
function flow

CMD
command trace

PAR
parser details

INT
program initialization and termination

ACC
access control (logging in)

UTL
utility functions

FSC(1)
file services -- level 1

FSC(2)
file services -- level 2

FSC(3)
file services -- level 3

FSC(4)
file services -- level 4

FSC(5)
file services -- level 5

FSC(6)
file services -- level 6

FSC(7)
file services -- level 7

FSC(8)
file services -- level 8

SOC(1)
socket services -- level 1

SOC(2)
socket services -- level 2

SOC(3)
socket services -- level 3

SOC(4)
socket services -- level 4

SOC(5)
socket services -- level 5

SOC(6)
socket services -- level 6

SOC(7)
socket services -- level 7

SOC(8)
socket services -- level 8

SQL
SQL processing

If no trace is active, NONE is displayed.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

None

Module

EZAFTPCL

Procedure name

debug

EZA2926I **The subcommand is not valid for the *mechanism* security mechanism**

Explanation

The user entered an FTP subcommand during a session that was protected with a security mechanism. The subcommand is not valid while the session is secured.

In the message text:

mechanism
The security mechanism used to secure the session. Valid values are TLS and GSSAPI.

System action

The command is ignored.

Operator response

No action is needed.

System programmer response

When the *mechanism* value is GSSAPI, secure the session using a different security mechanism, do not secure the session, or instruct the user to refrain from entering the subcommand.

When the *mechanism* value is TLS, more subcommands are permitted during a TLS-secured session when the TLSRFCLEVEL parameter value is RFC4217 than when the TLSRFCLEVEL parameter value is DRAFT. If the TLSRFCLEVEL parameter value is DRAFT, consider setting the TLSRFCLEVEL parameter value to RFC4217.

For more information, see the following sources:

- RFC 4217, *On Securing FTP with TLS*. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.
- The information about the steps for customizing the FTP client for TLS in z/OS Communications Server: IP Configuration Guide.
- The information about the TLSRFCLEVEL (FTP client and server) in z/OS Communications Server: IP Configuration Reference.
- The information about using security mechanisms in z/OS Communications Server: IP User's Guide and Commands.
- The information about the Locsite subcommand in z/OS Communications Server: IP User's Guide and Commands.

If the value TLSRFCLEVEL RFC4217 is not acceptable, or if using that value does not solve the problem, do one of the following:

- Do not secure the session
- Secure the session with a different mechanism
- Instruct the user to refrain from using the subcommand

User response

When the *mechanism* value is GSSAPI, either do not enter the subcommand, or report the error to the system programmer.

When the *mechanism* value is TLS, use the **LOCSTat** subcommand to display the TLSRFCLEVEL configuration option value, and report the error to the system programmer.

Problem determination

If the FTP server is z/OS V1R9 or later, you can display the server's TLSRFCLEVEL value with the **stat** subcommand. The server must be configured to use the value TLSRFCLEVEL RFC4217 to allow the client the full range of RFC 4217 functionality.

Source

z/OS Communications Server TCP/IP: FTP

Module

Not applicable.

Routing code

10

Descriptor code

12

Example

```
Command: safe
EZA2926I The subcommand is not valid for the TLS security mechanism.
```

EZA2927I SJISKANJI (NOTYPE requires type A (ASCII)

Explanation

An SJISKANJI subcommand was entered with a parameter of (NOTYPE. This parameter requires that the server be set to ASCII. For more information about the SJISKANJI command, see the z/OS Communications Server: IP User's Guide and Commands.

System action

The request is rejected. FTP continues.

Operator response

Issue the Type A command prior to entering the SJISKANJI (NOTYPE command.

System programmer response

No action is needed.

Module

EZAFTPCT

Procedure name

cliDBOpt, sjisk

EZA2928I **File contains binary data - enter binary subcommand before entering transfer subcommand**

Explanation

FTP has determined that the local file is tagged as a binary file. FTP will transfer this file only if the data type is set to binary.

System action

FTP continues.

Operator response

Enter the binary or TYPE I subcommand and reenter the transfer request.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

hfs_sndFile

EZA2929I **File contains ASCII data - enter ASCII subcommand before entering transfer subcommand**

Explanation

FTP has determined that the local file is tagged as an ASCII file. FTP will transfer this file only if the data type is set to ASCII.

System action

FTP continues.

Operator response

Enter the ASCII or TYPE A subcommand and reenter the transfer request.

System programmer response

No action is needed.

Module

EZAFTPCP.

Procedure name

hfs_sndFile

EZA2930I **Transfer failed because data cannot be translated**

Explanation

The data cannot be translated using the current translate table. The translate table was built from the code sets that are specified by the SBDATACONN parameter on LOCSITE or an SBDATACONN statement in the FTP.DATA file. At the time the table was built, one or more of the codepoints was reported as untranslatable for specific data codepoints. The data in this transfer has one or more of the codepoints.

System action

FTP continues.

Operator response

Use LOCSITE SBDATACONN to build or select a translation table that can translate the codepoints in the data.

System programmer response

No action is needed.

Module

EZAFTPCG, EZAFTPCP

Procedure name

hfs_rcvFile, sndFile

EZA2937I **SBDataconn codeset names: *codeset_1*, *codeset_2***

Explanation

This message is issued in response to the LOCSTAT subcommand.

codeset_1 and *codeset_2* are the names of the code sets used to define the data connection translate table specified by the SBDATACONN statement in the FTP.DATA file or by the LOCSITE SBDATACONN subcommand.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat

EZA2938I **Tagged *type* file translated with table built using file system *codepage_1*, network transfer *codepage_2***

Explanation

The FTP client detected that the local file is tagged with a coded character set ID (ccsid). A translation table is built for this data transfer using the codepage that corresponds to the ccsid.

type is either ASCII or EBCDIC.

codepage_1 is the codepage of the file.

codepage_2 is the codepage that was specified for the network transfer codepage on LOCSITE SBDATACONN or on the SBDATACONN statement in the FTP.DATA file.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

hfs_sndFile

EZA2939I **Tagged *type* file translated with current data connection translation table**

Explanation

The FTP client detected that the local file is tagged with a coded character set ID (ccsid). The data connection translate table is not defined with the SBDATACONN=(file_system_cp,network_transfer_cp) method. Without a network transfer codepage, the client cannot build a translate table using the ccsid of the file. The FTP client uses the translate table defined for the data connection.

type is either ASCII or EBCDIC.

System action

FTP continues.

Operator response

If you do not want to use the current data connection translate table, issue a LOCSITE SBDATACONN=(file_system_cp,network_transfer_cp) subcommand to provide a codepage to be used to build a translate table.

System programmer response

No action is needed.

Module

EZAFTPCP

Procedure name

hfs_sndFile

EZA2943I **SBSUB is set to *sbsub***

Explanation

SBSUB can be set to either TRUE or FALSE.

sbsub is one of the following:

TRUE
A substitution character is used when untranslatable characters are encountered during the data transfer.

FALSE
The data transfer will fail because of an untranslatable character. FALSE is the default setting.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2944I **SBSUBCHAR is set to *sbsubchar***

Explanation

During the data transfer, untranslatable characters will be replaced by the substitution character, *sbsubchar*. If SPACE is specified, then the ASCII or EBCDIC representation of a blank will be substituted. The SBSUBCHAR setting is ignored when SBSUB is set to FALSE.

sbsubchar is either SPACE or a substitution character in hexadecimal.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZAFTPCL

Procedure name

locstat()

EZA2946I **SBSUBChar parameter *sbsubchar* is not valid - SBSUBChar ignored**

Explanation

The SBSUBChar parameter must be either a hexadecimal number or SPACE.

sbsubchar is not a valid substitution character.

System action

The parameter is ignored.

Operator response

Reissue the LOCSITE command with a valid value specified for the SBSUBChar parameter. See the z/OS Communications Server: IP User's Guide and Commands, in the chapter about transferring data using the file transfer protocol for information about the parameters of the LOCSITE subcommand.

System programmer response

No action is needed.

Module

EZAFTPCK

Procedure name

locsite

EZA2947I **One or more characters were substituted during the transfer**

Explanation

During a data transfer, one or more untranslatable characters were found and were replaced by the substitution character (SBSUBCHAR). For single byte transfers the substitution character specified by SBSUBCHAR will be used. See the z/OS Communications Server: IP Configuration Reference and the z/OS Communications Server: IP User's Guide and Commands for information about specifying and displaying the SBSUBCHAR value.

System action

FTP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

ezaftpcg, ezaftpcp

Procedure name

hfs_rcvFile(), sndFile()

EZA2950I **usage: uuid@gen {-[pcPC]}**

Explanation

This message displays the correct format and syntax for the UUID command. For more information about the UUID command, see the *Network Computing System Reference Manual*.

System action

TCPIP continues.

Operator response

Correct the format and syntax as indicated and resubmit the command.

System programmer response

Assist the user as necessary.

Module

UUID@GEN

Procedure name

main

EZA2956E **GLBD: unable to listen on *type* address family**

Explanation

The Global Location Broker Daemon (GLBD), which controls the Global Location Broker database, which helps clients locate servers on the network, is unable to listen on a socket. The Berkeley socket concept is used for the Network Computing System (NCS), of which the GLBD is a part. Sockets are identified by a socket address composed of an address family, which defines the structure of the address, network address (host address), and port number (endpoint address). This message indicates that the GLBD cannot find a socket address for the indicated address family. No requests can be received over the socket.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the physical connections between the host and the network. Check to make sure that the indicated socket is available to the GLBD. For more information about the GLBD, see the z/OS Communications Server: IP Configuration Reference.

Module

GLBD

Procedure name

register_server

EZA2957E **GLBD: unable to listen on any address families**

Explanation

The remote procedure call to the Global Location Broker Daemon (GLBD) did not include any active address families. The Berkeley socket concept is used for the Network Computing System, of which the GLBD is a part. It can listen to more than one socket identified by a socket address composed of an address family, which defines the structure of the address, the network (host) address, and port number (endpoint address). Without an active address family, the GLBD is unable to receive requests over a connection.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the remote procedure call used to invoke the GLBD to make sure that it contains active address families. For more information about the GLBD, see the z/OS Communications Server: IP Configuration Reference.

Module

GLBD

Procedure name

main

EZA2961I **LLBD: (find_socket) called with no ready sockets**

Explanation

The procedure find_socket, which finds an available socket for a Local Location Broker Daemon (LLBD) connection, was called, but found no sockets ready to open a connection. A connection is not established.

System action

TCPIP continues.

Operator response

Resubmit the request when a socket is available.

System programmer response

Assist the user as necessary.

Module

LLBD

Procedure name

find_socket

EZA2962I ***info name* [*port*]**

Explanation

This message provides information about the use of a port.

Processing Request port
The indicated port has been used to submit a request that is now being processed.

Listening On port
NCS is listening on the indicated port for user requests.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LLBD

Procedure name

print_socket_info

EZA2963I **LLBD: (use_family) Can't create socket**

Explanation

The function use_family, which is called to set up a socket for a connection, was unable to create a socket. No connection is established.

System action

TCPIP continues.

Operator response

Resubmit the command when storage is available to create the socket.

System programmer response

Assist the user as necessary.

Module

LLBD

Procedure name

use_family

EZA2964I **LLBD: (use_family) Can't bind socket**

Explanation

The function use_family, which is called to set up a socket for a connection to Local Location Broker Daemon (LLBD), was unable to bind the socket because the bind request was rejected by the remote host. No connection is established.

System action

TCPIP continues.

Operator response

Resubmit the command when a socket is available on the remote host.

System programmer response

Check the remote host to determine why it is rejecting the LLBD BIND request.

Module

LLBD

Procedure name

use_family

EZA2965I **LLBD: Unable to obtain *type* family port**

Explanation

The Local Location Broker Daemon (LLBD) was unable to obtain a port for the indicated address family. The LLBD uses the Berkeley socket concept, allowing it to listen on multiple sockets, each of which has an address family, which defines the structure of the address, a network (host) address, and a port address. This message indicates the address family type for which the LLBD is unable to get a port. No connection is established.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the host to determine why no port can be allocated.

Module

LLBD

Procedure name

open_sockets

EZA2966E **LLBD: Unable to obtain any sockets**

Explanation

The Local Location Broker Daemon (LLBD) is unable to obtain any sockets for communication with the network. The LLBD uses sockets to listen for requests and to send replies.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Make sure that the LINK and DEVICE statements for the LLBD are configured properly and that the socket allocation requests submitted by LLBD specify valid, available sockets. Check the physical connections between the host and the network. Make sure that one or more ports are available to handle the sockets. For more information about socket errors, see the z/OS Communications Server: IP Programmer's Guide and Reference.

Module

LLBD

Procedure name

main

EZA2970I **usage lb@admin -n[q]**

Explanation

This message displays the correct format and syntax for the lb@admin command, which is used to invoke the location broker.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2971E **Unknown command: *command***

Explanation

The indicated command is not supported by the location broker. The command is not processed.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command. For more information about commands supported by the location broker, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2972E **Unknown option '*option*'**

Explanation

The indicated option is not supported by the location broker. The command is not processed.

System action

TCPIP continues.

Operator response

Correct the option and resubmit the command. For more information about options supported by the location broker, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2973I **Usage: unregister OBJ TYPE INTERFACE LOCATION**

Explanation

This message gives the format and syntax for the unregister command, which is used to remove an identifier from the location broker database. For more information about the unregister command, see the *Network Computing System Reference Manual*. The command is not processed.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

process_register_parms

EZA2974I **Usage: register OBJ TYPE INTERFACE LOCATION ANNOTATION {global | local}**

Explanation

This message displays the format and syntax for the register command, which is used to add an identifier to the location broker database. The command is not processed. For more information about the register command, see the *Network Computing System Reference Manual*.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2975I **Usage: set_broker {local | global} LOCATION**

Explanation

This message displays the format and syntax for the set_broker command, which is used to start the Location Broker Daemon. The command is not processed. For more information about the set_broker command, see the *Network Computing System Reference Manual*.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2976I **Usage: set_broker {local | global} LOCATION**

Explanation

This message displays the format and syntax for the set_broker command, which is used to start the Location Broker Daemon. The command is not processed. For more information about the set_broker command, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2977E ***specification*** **- illegal broker specification**

Explanation

The set_broker command contained an incorrect broker specification. The command is not processed. For more information about the specifications for set_broker command, see the *Network Computing System Reference Manual*.

System action

TCPIP continues.

Operator response

Correct the specification and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2948I **Usage: set_broker local | global**

Explanation

This message displays the format and syntax for the use_broker command, which is used to specify the location broker to be used. For more information about the use_broker command, see the *Network Computing System Reference Manual*.

System action

TCPIP continues.

Operator response

Correct the syntax and resubmit the command.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2979I **No entries match.**

Explanation

No entries were found matching the location broker found in the use_broker command. The command is not processed.

System action

TCPIP continues.

Operator response

Resubmit the use_broker command, specifying a valid location broker. For more information about the use_broker command, see the *Network Computing System Reference Manual*.

System programmer response

Assist the user as necessary.

Module

LB@ADMIN

Procedure name

main

EZA2991I **id=*id***

EZA2992I **Bindings for id @ *hash_table_position***

Explanation

These messages display information from the server's name table in a readable form.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

NAMETBL

Procedure name

NAMETBL_dump_tab

EZA2994I **Unexpected (or unimplemented) type (“**

EZA2995I **) encountered in *label***

EZA2996I ; exiting.

Explanation

The function BACKEND_init, which initializes the code for the BACKEND procedure, encountered a type that it does not recognize in the indicated label. The function exits the label containing the unrecognized type.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

BACKEND

Procedure name

BACKEND_init

EZA2997I name *address* is "*name*"

Explanation

This message translates a name table entry to a readable name.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

BACKEND

Procedure name

NAMETABLE_id_to_string

EZA2998I NIDL compiler invoked

Explanation

This message indicates that the Network Interface Definition Language (NIDL) compiler has been invoked.

System action

The NIDL compiler is initialized. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MAIN

Procedure name

main

EZA2999I **NIDL compiler ended**

Explanation

The Network Interface Definition Language compiler (NIDL) has ended.

System action

The NIDL compiler ends. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MAIN

Procedure name

main

Chapter 5. EZA3*xxxx* messages

EZA3000I **unexpected EOF inside a comment at line *line***

Explanation

The Network Interface Definition Language (NIDL) compiler encountered an unexpected end of file in a comment at the indicated line in the current file. Processing of the file halts.

System action

TCPIP continues.

Operator response

Notify the system programmer of the problem.

System programmer response

Remove the end of file marker from the comment and recompile. For more information about NIDL, see the z/OS Communications Server: IP Programmer's Guide and Reference.

Module

LEX_YY

Procedure name

main

EZA3001E **Illegal option ftype in getflags: *ftype***

Explanation

The Network Interface Definition Language (NIDL) compiler encountered an incorrect option in the procedure getflags, which is used to set processing flags for the compiler. The option is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the indicated option and recompile. For more information about NIDL compiler options, see the z/OS Communications Server: IP Programmer's Guide and Reference.

Module

GETFLAGS

Procedure name

main

EZA3002E **Unknown flag (*flag*) in getflags**

Explanation

The Network Interface Definition Language (NIDL) compiler encountered an option that it does not recognize. The option is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the indicated option and recompile. For more information about NIDL compiler options, see the z/OS Communications Server: IP Programmer's Guide and Reference.

Module

GETFLAGS

Procedure name

main

EZA3003E **Options table:**

Explanation

This message precedes the printing of the options table for the Network Interface Definition Language (NIDL) compiler in response to an incorrect option.

System action

The options table is displayed. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the options as necessary and recompile.

Module

GETFLAGS

Procedure name

printflags

EZA3004E ***option***

Explanation

The indicated option for the Network Interface Definition Language (NIDL) compiler is incorrect. The compiler halts.

System action

The compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the option and recompile. For information on NIDL compiler options, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3005E **illegal option in printflags: *option***

Explanation

The indicated option for the Network Interface Definition Language (NIDL) compiler is either incorrect or not allowed.

System action

The compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the option and recompile. For more information about NIDL compiler options, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3006E ***flag***

Explanation

The indicated INTARG flag for the Network Interface Definition Language (NIDL) compiler is incorrect.

System action

The compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the flag and recompile. For more information about NIDL compiler flags, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3007E ***STRARG_flag***

Explanation

The indicated STRARG flag for the Network Interface Definition Language (NIDL) compiler is incorrect.

System action

The compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the flag and recompile. For more information about flags for the NIDL compiler, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3008E ***flag***

Explanation

The indicated Network Interface Definition Language (NIDL) flag, which is either a TOGGLEARG flag or an ASSERTARG flag, is incorrect.

System action

The NIDL compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the flag and recompile. For more information about NIDL compiler flags, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3009E ***flag***

Explanation

The DENYARG flag for the Network Interface Definition Language (NIDL) compiler is incorrect.

System action

The NIDL compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the DENYARG flag and recompile. For more information about NIDL compiler options, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3010E ***flag***

Explanation

The CHRARG flag for the Network Interface Definition Language (NIDL) compiler is incorrect.

System action

The NIDL compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the CHRARG flag and recompile. For more information about NIDL compiler options, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3011E ***flag***

Explanation

The FLTARG flag for the Network Interface Definition Language (NIDL) compiler is incorrect.

System action

The NIDL compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the FLTARG flag and recompile. For more information about NIDL compiler options, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3012E ***flag***

Explanation

The LONGARG flag for the Network Interface Definition Language (NIDL) compiler is incorrect.

System action

The NIDL compiler halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the LONGARG flag and recompile. For more information about NIDL compiler options, see the *Network Computing System Reference Manual*.

Module

GETFLAGS

Procedure name

printflags

EZA3013E **usage: NIDL fileName [flags]**

Explanation

The NIDL command, which is used to invoke the Network Interface Definition Language (NIDL) compiler, was incorrect. The command is not processed. This message gives the correct format and syntax for the NIDL command. For more information about the NIDL compiler, see the *Network Computing System Reference Manual*.

System action

TCPIP continues.

Operator response

Correct the NIDL command and resubmit it.

System programmer response

Assist the user as necessary.

Module

FRONTEND

Procedure name

explain_args

EZA3014E **-ansi is no longer needed.**

Explanation

The NIDL command, which is used to invoke the Network Interface Definition Language (NIDL) compiler, was submitted with the ansi option specified. The ansi option is no longer necessary. The command is not processed.

System action

TCPIP continues.

Operator response

Resubmit the command without specifying the ansi option.

System programmer response

Assist the user as necessary.

Module

FRONTEND

Procedure name

NIDL_init

EZA3030E **(alloc_socket) Can't create socket, errno=*errno***

Explanation

The procedure alloc_socket, which is called to allocate a socket, was unable to create a new socket.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@C

Procedure name

alloc_socket

EZA3031E **(alloc_socket) Can't bind socket, errno=*errno***

Explanation

The function alloc_socket, which is called to allocate a socket, was unable to bind a socket to open communications.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@C

Procedure name

alloc_socket

EZA3032E **(send_pkt) cannot enable broadcast, status=*status***

Explanation

The function send_pkt, which is called to send packets to clients, is unable to broadcast a packet over the network because the socket from which the packet is to be sent does not have the correct status.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Determine the reason the socket has an incorrect status and respond as indicated.

Module

RPC@C

Procedure name

send_pkt

EZA3033E **(send_pkt) cannot enable broadcast, errno=*errno***

Explanation

The function send_pkt, which is called to send a packet, could not send a broadcast message. No packet is sent.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@C

Procedure name

send_pkt

EZA3034E **(recv_pkt/client) Rcvd packet from wrong place; wanted “*address*”**

EZA3035E **(recv_pkt/client) got “*address*”**

Explanation

The function recv_pkt/client, which is called to receive a packet, received a packet from an address other than the one it expected. The packet is discarded.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the client and both indicated addresses to determine why the client is receiving packets from an incorrect address.

Module

RPC@C

Procedure name

recv_pkt/client

EZA3036I **(recv_pkt/client) Rcvd callback (ptype=*packet_type*)**

Explanation

The function recv_pkt/client, which is called to receive a packet, received a callback from a client indicating that the last packet sent to that client has been received.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

recv_pkt/client

EZA3037I **(recv_pkt/client) Rcvd packet not for my activity**

Explanation

The host received a packet that does not match the host's current activity. If the sequence number of the packet is correct, the packet will be acknowledged, preventing retransmission of the packet. Otherwise the packet will be discarded, causing the client to resend the packet when the acknowledgment time out expires.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

recv_pkt/client

EZA3038I **(recv_pkt/client) Rcvd packet with bad seq (should be *number*, is *number* ptype=*packet_type*)**

Explanation

The host has received a packet with an incorrect sequence number. The packet is discarded without acknowledgment, causing the client to resend the packet after the acknowledgment time out has expired. The actual sequence number, expected sequence number, and packet type are displayed in this message.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

recv_pkt/client

EZA3039I **(ack_replies) Acking [*address*, *sequence_number*]**

Explanation

The function ack_replies, which is called to send acknowledgments for packets that have been received, is sending an acknowledgment to the indicated address for the packet with the indicated sequence number.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

ack_replies

EZA3040E **(ack_replies) Can't send ACK, errno=*errno***

Explanation

The function ack_replies, which is called to acknowledge packets that have been received, is unable to send an acknowledgment.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* given in this message to determine the cause of the error and respond as indicated.

Module

RPC@C

Procedure name

ack_replies

EZA3041I **(await_reply) timeout**

Explanation

The host is currently in a time out initiated by the procedure await_reply, which is called to put the host in a wait state to receive packets.

System action

After the timeout, the host resumes processing. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

await_reply

EZA3042I **(await_reply) timeout (EINTR)**

Explanation

A request from the function await_reply, which waits for a reply to a packet, to the operating system was interrupted. The request will be resubmitted to the system.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

await_reply

EZA3043I **(rpc_sar) Starting to ping**

Explanation

The host is sending a PING packet to check for the existence of other hosts on the network. For more information about the PING function, see the z/OS Communications Server: IP System Administrator's Commands.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

rpc__sar

EZA3044I **(rpc_sar) Too many pings...signaling**

Explanation

The local host has sent the maximum number of PING packets to check for the existence of another host on the network. The local host assumes that the host it is seeking does not exist. No more PING packets are sent.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

rpc__sar

EZA3045I **(rpc__sar) Rcvd “working” during frag send!**

Explanation

The host received a “working” packet, but the host is sending fragmented packets, disrupting data transmission. The packet is acknowledged.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

rpc__sar

EZA3046I **(rpc__sar) Rcvd “working” pkt; seq=*number***

Explanation

The host has received a “working” packet with the indicated sequence number. The packet is acknowledged.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

rpc__sar

EZA3047I **(rpc__sar) Got “fack” for non-large request!**

Explanation

The host received a fragmentation acknowledgment packet for a request that was not large enough to need fragmentation. This indicates that the client is configured to receive a smaller maximum transmission unit (MTU) than the host is configured to send. The packet is transmitted in fragments to allow the client to receive it.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the client to determine why it is acknowledging fragmentation when none is necessary.

Module

RPC@C

Procedure name

rpc__sar

EZA3048I **(rpc__sar) Got “response” during frag recv!**

Explanation

The host received a response packet while sending a fragmented packet. The response packet is received and acknowledged after sending of the fragmented packet is completed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

rpc__sar

EZA3049I **(rpc__sar) Got non-frag “response” during frag send!**

Explanation

The host received a non-fragmented response packet while sending a fragmented packet. The response will be accepted and acknowledged after the host is through sending the fragmented packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@C

Procedure name

rpc__sar

EZA3050E (use_family) Can't create socket

Explanation

The function use_family, which is used to create and register a socket for listening, was unable to create a socket. This prevents the host from receiving requests over the connection.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the host to determine why it is unable to create a socket, and respond as indicated. For more information about socket errors, see the z/OS Communications Server: IP Programmer's Guide and Reference.

Module

RPC@S

Procedure name

use_family

EZA3051E (use_family) Can't bind socket, errno=*errno*

Explanation

The function use_family, which is used to create and register a socket for listening, could not bind a socket to open a connection, preventing the host from receiving requests over that connection.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@S

Procedure name

use_family

EZA3052E **(rpc__use_family) Can't getsockname, errno=*errno***

Explanation

The function getsockname, which is used by the function rpc__use_family to get the name of a socket to be used for listening, was unsuccessful. The connection is not established.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@S

Procedure name

use_family

EZA3053E **(rpc__use_family) Can't get my netaddr**

Explanation

The function rpc__use_family, which uses a socket to listen for requests from clients, was unable to get the network address for the socket. The socket is closed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the host to determine why the network address was not passed to rpc__use_family. For more information about this function, see the *Network Computing System Reference Manual*.

Module

RPC@S

Procedure name

rpc__use_family

EZA3054E **(rpc__use_family) Can't get socket**

Explanation

The function rpc__use_family, which uses a socket to listen for requests, was unable to get a socket. The connection is not established.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the host to determine why no socket is available and respond as indicated. For more information about socket errors, see the z/OS Communications Server: IP Programmer's Guide and Reference.

Module

RPC@S

Procedure name

rpc__use_family

EZA3055I **(scan_activities) Freeing passive *connection***

Explanation

The function scan_activities, which retransmits unacknowledged replies and frees old activities, is freeing the indicated passive connection.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

scan_activities

EZA3056I **(scan_activities) Passivating idle *connection***

Explanation

The function scan_activities, which retransmits unacknowledged replies and frees old activities, is changing the state of the indicated connection from idle to passive open.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

scan_activities

EZA3057I **(scan_activities) Dropping reply [*identifier*, *sequence_number*]**

Explanation

The function scan_activities, which retransmits unacknowledged replies and frees old activities, is dropping the indicated packet because it has been sent the maximum allowable number of times.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

scan_activities

EZA3058I **(scan_activities) Retransmitting reply (state=*state*) [*id*, *reply_packet_sequence_header_*]**

Explanation

The function scan_activities, which retransmits unacknowledged replies and frees old activities, is resending the indicated packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

scan_activities

EZA3059I **(ping_common) Working (ptype=*packet_type*) [*id*, *sequence_number*]**

Explanation

The function ping_common, which is used to process request and PING packets whose sequence numbers indicated that they are current, is processing the indicated packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

ping_common

EZA3060I **(ping_common) Resending reply (state=*state*, ptype=*packet_type*, frag=*fragmentation_number*) [*id*, *sequence_number*]**

Explanation

The function ping_common, which is used to process request and PING packets whose sequence number indicates that they are current, is resending the indicated reply packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

ping_common

EZA3061I **(ping_common) No call (state=*state*, ptype=*packet_type*) [*id.*, *sequence_number*]**

Explanation

The function ping_common, which is used to process request and PING packets whose sequence numbers indicate that they are current, received a packet that did not contain a call for a service.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

ping_common

EZA3062I **(who_are_you) Doing callback**

Explanation

The function who_are_you, which is used to determine the identity of a client, is doing a callback to the client.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

who_are_you

EZA3063I **(who_are_you) Can't bind to client, st=*status***

Explanation

The function who_are_you, which is used to determine the identity of a client, is unable to bind a socket for communication with the client. The reason is indicated by the *status* displayed in this message.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the *status* displayed in this message.

Module

RPC@S

Procedure name

who_are_you

EZA3064I **(who_are_you) fault, st=*status***

Explanation

The function who_are_you, which is used to determine the identity of a client, has encountered an error. The connection to the client is not established. The reason is indicated by the *status* displayed in this message.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the *status* displayed in this message.

Module

RPC@S

Procedure name

who_are_you

EZA3065I **(who_are_you) who_are_you failed, st=*status***

Explanation

The function who_are_you, which is used to determine the identity of a client, was unsuccessful. The reason is indicated by the *status* displayed in this message. The connection is closed and the function rpc__free_handle is called to free the socket.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the *status* displayed in this message.

Module

RPC@S

Procedure name

who_are_you

EZA3066I **(do_request_frag) Bad initial frag**

Explanation

The function do_request_frag, which is called to handle fragmented request packets, received an initial fragment other than the one that was expected.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_request_frag

EZA3067I **(do_request_frag) Out of order frag; expecting *number* [*packet_type*, *sequence_number*]**

Explanation

The function do_request_frag, which is used to handle fragmented request packets, received a fragment other than the one that is expected. The fragment is not acknowledged, prompting the sending client to resend the fragment after the acknowledgment timeout.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_request_frag

EZA3068I **(do_request) Unknown interface [*packet_type*, *sequence_number*]**

Explanation

The function do_request, which accepts and handles client request packets, received a packet from an interface that is not registered with a location broker. The packet is refused and the request is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Register the interface with the location broker and resubmit the request. For more information about registering an interface with a location broker, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Module

RPC@S

Procedure name

do_request.

EZA3069I **(do_request) Opnum out of range [*id*, *sequence_number*]**

Explanation

The function do_request, which accepts and handles client request packets, received a packet whose operation number was outside the valid range. The packet is rejected, prompting the sending client to create a new packet to send the request.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_request.

EZA3070I **(do_request) Got inappropriate request (state=*state*) [*id*, *sequence_number*]**

Explanation

The function do_request, which accepts and handles client requests, received a request packet during the processing of a duplicate request. Processing of the old request halts. The new request is returned, prompting the client to retransmit the request and allowing the host time to shut down processing of the old request.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_request.

EZA3071I **(do_request) Old sequence, previous=*sequence_number* [*id*, *sequence_number*]**

Explanation

The function do_request, which accepts and handles client request packets, received a packet whose sequence number indicates that it is not current. The packet is rejected.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_request

EZA3072I **(do_request) Mismatch sequence during frag reassembly [*id*, *sequence_number*]**

Explanation

The function do_request, which accepts and handles client request packets, received a packet with an incorrect sequence number during the reassembly of a fragmented packet. The fragment is rejected. The fragmented packet is reassembled.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_request.

EZA3073E **(do_request) Protocol error [*id*, *sequence_number*]**

Explanation

The function do_request, which accepts and handles client request packets, encountered a protocol error caused by an out of sequence packet. The packet is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the sending device to determine why it is sending out of sequence packets.

Module

RPC@S

Procedure name

do_request.

EZA3074E **(do_request) Fault while executing request, st=*status***

Explanation

The function do_request, which accepts and handles client request packets, encountered an error while processing a request. The reason is indicated by the *status* displayed in this message.

System action

Processing halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Respond as indicated by the *status* portion of this message.

Module

RPC@S

Procedure name

do_request.

EZA3075I **(do_ping) No call (no activity for this call) [*id, sequence_number*]**

Explanation

The function do_ping, which handles PING packets, received a packet that did not contain a request for an activity.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_ping

EZA3076I **(do_ping) No call)higher numbered ping), ahint=*activity_number*, previous=*old*us.sequence_number* [*id*, *sequence_number*]**

Explanation

The function do_ping, which handles PING packets, received a packet with a sequence number higher than the sequence number of the current activity, indicating that the “quit” packet for that activity has been lost. The current activity is stopped.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_ping

EZA3077I **(do_ping) Drop ping [*id*, *sequence_number*]**

Explanation

The function do_ping, which is called to handle PING packets, is dropping the indicated PING packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_ping

EZA3078I **(do_ack) No or incorrect ahint in ack, ahint=*activity_number* [*id*, *sequence_number*]**

Explanation

The function do_ack, which handles acknowledgment packets, received a packet with a missing or incorrect activity number. The packet is dropped, then resent.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_ack.

EZA3079I **(do_fack) No or incorrect ahint in fack, ahint=*activity_number* [*id*, *sequence_number*]**

Explanation

The function do_fack, which handles fragmented acknowledgment packets, received a packet with a missing or incorrect activity number for a fragmented packet. The packet is dropped, the packet is resent.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_ack.

EZA3080I **(do_fack) Anomalous fack, state=*state*, prev_seq=*sequence_number*, frag=*fragment_number*, [*id*, *sequence_number*]**

Explanation

The function do_fack, which handles fragmented acknowledgment packets, received a packet with a state, sequence number, or fragmentation number other than what was expected.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_fack

EZA3081I **(do_quit) No call (no activity for this call) [*id*, *sequence_number*]**

Explanation

The function do_quit, which handles quit packets, received a packet that did not contain an activity number.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_quit

EZA3082I (do_bad_pkt) Bad pkt, ptype=*packet_type* [*id*, *sequence_number*]

Explanation

The function do_bad_pkt, which handles incorrect packets, has been called for the indicated packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

do_bad_pkt

EZA3083I (rpc__listen_dispatch) Trash ahint (*activity_number*)

Explanation

The function rpc__listen_dispatch, which does server-side processing for one incoming packet, received a packet with an activity number that is greater than the range of valid activity numbers. The packet is discarded.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

rpc__listen_dispatch

EZA3084I **(rpc__listen_dispatch) Trashi ihint (*activity_number*)**

Explanation

The function rpc__listen_dispatch, which does server-side processing for one packet, received a packet with an IP address outside the valid range. The packet is discarded.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

rpc__listen_dispatch

EZA3085I **(rpc__listen_dispatch) Server boot time mismatch**

Explanation

The function rpc__listen_dispatch, which handles server-side processing for one packet at a time, received a packet with a server boot time field that does not match the current server. This indicates that the server has been rebooted since the last communication with this client. The packet is rejected, prompting the client to update their tables with the correct boot time. The packet will be resent by the client.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

rpc__listen_dispatch

EZA3086I **(check_for_pkt) Rcvd (ptype=** ***type*****) [*****id*****,** ***sequence_number*****]**

Explanation

The function check_for_pkt, which checks all sockets for data and processes any non-request packets, received the indicated packet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

check_for_pkt

EZA3087I **(check_for_pkt) Waiting...**

Explanation

The function check_for_pkt, which checks all sockets for data and processes any non-request packets, is currently in a waiting state.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

check_for_pkt

EZA3088I **(check_for_pkt) Checking for pkts...**

Explanation

The function check_for_pkt, which checks all sockets for data and processes any non-request packets, is checking all sockets for data.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

check_for_pkt

EZA3089I **(listen) task *task_number***

Explanation

The function listen, which listens for requests from clients, received a packet with a request for the indicated task.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

listen

EZA3090I **(listen) Spurious quit activity fault ignored**

Explanation

The function listen, which listens for requests from clients, received a quit request for an activity that has already been stopped. The request is ignored.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

listen

EZA3091I **(listen) wait failed *status***

Explanation

The function listen, which listens for requests from clients, did not receive data after entering a wait state to wait for data.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

listen

EZA3092I **(listen) Spurious quit activity fault ignored**

Explanation

The function listen, which listens for requests from clients, received a quit request for an activity that has already been stopped. The request is ignored.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@S

Procedure name

listen

EZA3093E **(rpc) *** FATAL ERROR "*error*" at *file\line* *****

Explanation

The server has encountered an error in the indicated data set at the indicated line.

System action

Processing of the data set halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the data set as indicated and resubmit it for processing.

Module

RPC@UTIL

Procedure name

rpc__die

EZA3094E **(rpc__lock) Can't get *lock* lock at *file\line***

Explanation

The function rpc__lock, which is used to make sure that only one processing thread is occurring during many RPC runtime routines, was unable to lock the indicated data set at the indicated line.

System action

Processing of the data set halts. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Correct the indicated line in the data set and resubmit the data set for processing.

Module

RPC@UTIL

Procedure name

rpc__lock

EZA3095E **(rpc__recvfrom) Error in recv, errno=*errno***

Explanation

The function rpc__recvfrom, which is used to receive a packet from a socket, was unable to receive a packet.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@UTIL

Procedure name

rpc__recvfrom

EZA3096E **(rpc__recvfrom) Bad RPC version (*version*)**

Explanation

The function rpc__recvfrom, which receives a packet from a socket, received a packet sent by an incompatible RPC version. The packet is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Replace the incompatible RPC version with a compatible one. For more information about RPC versions, see the *Network Computing System Reference Manual*.

Module

RPC@UTIL

Procedure name

rpc__recvfrom

EZA3097E **(rpc__recvfrom) Bad pkt type (*type*)**

Explanation

The function rpc__recvfrom, which receives a packet from a socket, received a packet with an incorrect type. The packet is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the sending device to determine why it is sending incorrect packet types.

Module

RPC@UTIL

Procedure name

rpc__recvfrom

EZA3098E **(rpc__recvfrom) Packet too short: is *length* data length is *length***

Explanation

The function rpc__recvfrom, which receives a packet from a socket, received a packet that is too small. The packet is not processed.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the sending device to determine why it is sending packets that are smaller than allowed.

Module

RPC@UTIL

Procedure name

rpc__recvfrom

EZA3099E **(rpc__sendto) sento failed, errno=*errno***

Explanation

The function rpc__sendto, which sends a packet to a socket, was unsuccessful.

errno is the UNIX System Services Return Code. These return codes are listed and described in the z/OS UNIX System Services Messages and Codes.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *errno* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@UTIL

Procedure name

rpc__sendto

EZA3100E **(periodic_task) "*task*" task exited, status=*status***

Explanation

A task that is called periodically has been exited, rather that entering a wait state until the next call. The *status* displayed in this message indicates the reason.

System action

The task is exited. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the *status* displayed in this message to determine the cause of the error and respond as indicated.

Module

RPC@UTIL

Procedure name

periodic_task

EZA3101I **rpc_vers: *version***

EZA3102I **xptype: *packet_type* (*type*)**

EZA3103I **flags: *header_flags***

EZA3104I **int_rep: *integer_representation***

EZA3105I **char_rep: *character_representation***

EZA3106I **float_rep: *floating_decimal_representation***

Explanation

These messages display information about the RPC version being used.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@UTIL

Procedure name

rpc__dump_pkt_hdr

EZA3107I **object uuid: *object_identifier***

EZA3108I **interface uuid: *interface_identifier***

EZA3109I **activity uuid: *activity_identifier***

EZA3110I **server boot time: *time***

EZA3111I **interface version: *version***

EZA3112I **sequence number: *number***

EZA3113I **operation number: *number***

EZA3114I **interface hint: *number***

EZA3115I **activity hint: *number***

EZA3116I **fragment #: *number***

EZA3117I **length of body: *length***

Explanation

These messages display information about a packet received by the RPC utility.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPC@UTIL

Procedure name

rpc__dump_pkt_hdr

Chapter 6. EZA4*xxxx* messages

EZA4250E **couldn't do udp_create**

Explanation

The svcudp_create() call which creates a UDP-based service transport to which it returns a pointer, was unsuccessful.

System action

Portmapper stops.

Operator response

Inform the system programmer of this message.

System programmer response

Check for storage availability, check that all the udp_create parameters have been correctly specified and restart the program.

Module

PORTMAP

Procedure name

Main()

EZA4251E **couldn't do tcp_create**

Explanation

The svctcp_create() call, which creates a TCP-based service transport to which it returns a pointer, was unsuccessful.

System action

Portmapper stops.

Operator response

Inform the system programmer of this message.

System programmer response

Check for storage availability, check that all the tcp_create parameters have been correctly specified, and restart the program.

Module

PORTMAP

Procedure name

Main()

EZA4252I **portmap: started**

Explanation

The portmapper process has been initiated. Portmapper is the software that supplies client programs with the port numbers of server programs.

System action

Portmapper continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

PORTMAP

Procedure name

reg_portmapper

EZA4253E **svc_run returned unexpectedly**

Explanation

The procedure svc_run, which accepts RPC requests and calls the appropriate service using svc_getreq(), was unsuccessful.

System action

Portmapper stops.

Operator response

Inform the system programmer of this message.

System programmer response

Check for the correct PORTMAP specifications in the *hlq*.PROFILE.TCPIP configuration data set, restart PORTMAP, and run the program again. If the error persists, call the IBM Software Support Center.

Module

PORTMAP

Procedure name

svc_run

EZA4254I **portmap: procedure *procedure* called from *address***

Explanation

Indicates the procedure that has been called and the address where the call was issued. This message is written to an output file.

System action

Portmapper continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

PORTMAP

Procedure name

reg_srv

EZA4255E **svc_sendreply**

Explanation

The procedure svc_sendreplay(), which is called by the service dispatch routine to send the results of the call to the caller, was unsuccessful.

System action

Portmapper stops.

Operator response

Inform the system programmer of this message.

System programmer response

Check for the correct PORTMAP specifications in the *hlq*.PROFILE.TCPIP configuration data set, restart PORTMAP, and run the program again. If the error persists, call the IBM Software Support Center.

Module

PORTMAP

Procedure name

reg_srv

EZA4256I **PORTMAPPER STARTED**

Explanation

The UNIX System Services Portmapper issues this message when it has completed initialization and is ready to accept requests.

System action

Portmapper continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

portmap

Procedure name

main(), svc_run()

EZA4257I ***application* STOPPED**

Explanation

The UNIX System Services application specified by the *application* value issues this message just before it exits in response to a **`stop`** command from the operator.

In the message text:

application
The name of the application specified.

System action

The *application* stops.

Operator response

No action is needed.

System programmer response

Restart the *application* as needed.

Module

portmap or rpcbind

Procedure name

main()

EZA4258I **RPCBIND INITIALIZATION COMPLETED AT *time* ON *date***

Explanation

The UNIX System Services rpcbind server issues this message when it has completed initialization and is ready to accept requests.

In the message text:

time
The time that the initialization was completed, in local time.

date
The date that the initialization was completed, in local time.

System action

rpcbind continues.

Operator response

If *time* and *date* are accurate in local time, no action is required. Otherwise, notify the system programmer.

System programmer response

If *time* and *date* do not reflect the current local time and date, correct the configuration and start rpcbind again. Local time is affected by time zone information such as:

- The TZ environment variable
- The _TZ environment variable
- The LC_TOD category

See the information about building a locale in z/OS XL C/C++ Programming Guide for a description of LC_TOD, which is a nonstandard z/OS C/C++ proprietary locale category. See the information about setting your local time zone in z/OS UNIX System Services User's Guide, or the information about customizing a time zone in z/OS XL C/C++ Programming Guide for information about customizing the local time zone.

User response

No action required.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

oco part

Routing code

11

Descriptor code

0

Example

```
EZA4258I RPCBIND INITIALIZATION COMPLETED AT 16:15:48 ON  11/01/05.
```

EZA4259I **RPCBIND BIND() TO *stack protocol* PORT *portnum* FAILED ERRNO = *errno* ERRNO2 = *errnojr***

Explanation

The rpcbind server attempted to use the C/C++ bind() API using *protocol* to *stack* on *portnum*. The interface returned -1 with a decimal *errno* and hexadecimal *errnojr*. See the z/OS XL C/C++ Runtime Library Reference to determine the reason for the returned value of the bind() call. The most common returned value is 1115, EADDRINUSE. This means that another application is listening on the specified port. Use the Netstat command netstat -p *stack* -P *portnum* in the z/OS UNIX environment, or NETSTAT TCP *stack* (PORT *portnum* in the TSO environment to determine what application is listening on the specified port.

In the message text:

stack
: The name of the TCP/IP stack.

protocol
: The value is either TCP or UDP.

portnum
: The port number, usually 111, that RPCBIND tried to bind to a socket.

errno
: The UNIX System Services return code. These return codes are listed in and described in the return codes (errnos) chapter in z/OS UNIX System Services Messages and Codes. The most likely *errno* value for this error is 1115 for EADDRINUSE.

errnojr
: The hexadecimal UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the reason codes (errnojrs) chapter of the z/OS UNIX System Services Messages and Codes, where the reason codes are listed. The most likely *errnojr* value for this error is 0xXXXX7232, JrBindAddrInUsed.

System action

rpcbind terminates.

Operator response

Contact the system programmer.

System programmer response

If the *errno* is 1115 (EADDRINUSE), use the netstat -P *portnum* command to determine which application is using the specified port. End the application's use of the port and restart rpcbind. If the *errno* is not 1115, see z/OS UNIX System Services Messages and Codes to determine the reason for the returned values. Determine and correct the failure. Restart rpcbind.

If the *errno* is 1115 for EADDRINUSE, you can use the Netstat command's port report to determine which application is already using port 111. In the following example, the PORTMAP address space already has port 111 and needs to be stopped before starting RPCBIND.

```
netstat -p TCPCS  -P 111
MVS TCP/IP NETSTAT CS V1R8       TCPIP Name: TCPCS           09:06:27
User Id  Conn     State
-------  ----     -----
PORTMAP  000000A1 Listen
  Local Socket:   0.0.0.0.111
  Foreign Socket: 0.0.0.0.0
PORTMAP  000000A0 UDP
  Local Socket:   0.0.0.0.111
  Foreign Socket: *.*
```

User response

Not applicable.

Problem determination

Not applicable.

Source

MVS Platform

Module

rpcbind.c

Routing code

Not applicable.

Descriptor code

Not applicable.

Example

```
EZA4259I RPCBIND BIND() TO TCPCS    UDP PORT 111 FAILED ERRNO = 1115  ERRNO2 =  0X74637232
```

EZA4260I **UNSUPPORTED MODIFY COMMAND PARAMETER *text*.**

Explanation

The UNIX System Services rpcbind server issues this message when it receives a syntactically incorrect modify request from the MVS operator. A correct modify request is in the form TRACE=*keyword*. The valid keywords can be found in z/OS Communications Server: IP System Administrator's Commands.

In the message text:

text
The unrecognized text entered after the procedure name.

System action

rpcbind continues. The modify request is ignored.

Operator response

Contact the system programmer.

System programmer response

See the information about the MODIFY command -- RPCBIND in z/OS Communications Server: IP System Administrator's Commands to correct the syntax of the modify request.

User response

Not applicable.

Problem determination

Not applicable.

Source

Not applicable.

Module

rpcbind.c

Routing code

11

Descriptor code

0

Example

```
EZA4260I F RPCBIND,TRACE=XXXX EZA4260I UNSUPPORTED MODIFY COMMAND PARAMETER XXXX.
```

EZA4261I **RPCBIND KEYWORD *keyword* SET *status***

Explanation

The UNIX System Services rpcbind server issues this message in response to the MVS modify RPCBIND TRACE request. The message displays the setting of a TRACE keyword. See the information about the MODIFY command -- RPCBIND in z/OS Communications Server: IP System Administrator's Commands to set the keyword.

In the message text:

keyword
The type of trace. The values for keyword are FLOW, ERROR, LOG, XDR.

status
The value is either ON or OFF.

System action

rpcbind continues.

Operator response

Contact the system programmer.

System programmer response

Verify that the TRACE settings are set correctly. See the information about the MODIFY command -- RPCBIND in z/OS Communications Server: IP System Administrator's Commands to set the keyword.

User response

Not applicable.

Problem determination

Not applicable.

Source

MVS Platform

Module

iprpcoe/rpcbind/portmap.c

Routing code

11

Descriptor code

0

Example

```
f rpcbind,trace=?
EZA4261I RPCBIND KEYWORD FLOW SET ON
EZA4261I RPCBIND KEYWORD ERROR SET ON
EZA4261I RPCBIND KEYWORD LOG SET ON
EZA4261I RPCBIND KEYWORD XDR SET ON
```

EZA4262I THE RPCBIND PID FILE rpcbind.pid CANNOT BE WRITTEN TO DIRECTORY *dir* - *reason*

Explanation

The UNIX System Services rpcbind server issues this message when it tries to write the PID file, rpcbind.pid, to the specified directory. The directory might not exist or it might be mounted as read only.

In the message text:

dir
The directory specified with the rpcbind -i option, or the default directory /etc if no -i option was specified.

reason
The reason the PID file cannot be written to the directory specified by the *dir* value.

System action

rpcbind stops.

Operator response

Contact the system programmer.

System programmer response

If the specified reason is DIRECTORY DOES NOT EXIST, create the directory using the **`mkdir`** command with the specified directory name from the UNIX shell. If the specified reason is DIRECTORY IS READ ONLY, ensure that the file system on which the directory is mounted is read/write. Correct the problem indicated by *reason*, or use the -i option of the **`rpcbind`** command to specify a different directory.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

rpcbind.c

Routing code

11

Descriptor code

0

Example

```
EZA4262I THE RPCBIND PID CANNOT BE WRITTEN TO DIRECTORY '/XXX' - DIRECTORY DOES NOT EXIST
```

EZA4263I **RPCBIND COULD NOT BE RUN AS NONSWAPPABLE**

Explanation

The UNIX System Services rpcbind server issues this message when the -n option was specified and the user ID associated with rpcbind does not have READ access to resource BPX.STOR.SWAP in the FACILITY class.

System action

rpcbind stops.

Operator response

Contact the system programmer

System programmer response

Ensure that rpcbind has READ access to the BPX.STOR.SWAP resource in the FACILITY class before restarting rpcbind with the -n option.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

rpcbind.c

Routing code

11

Descriptor code

0

Example

```
EZA4263I RPCBIND COULD NOT BE RUN AS NONSWAPPABLE
```

EZA4264I **INCORRECT RPCBIND OPTION *option* SPECIFIED**

Explanation

The UNIX System Services rpcbind server issues this message when it receives an incorrect option. One possible reason for the error is that options are case sensitive even though messages that appear on the z/OS console are uppercase. Therefore, if an option appears to be correct in a message, verify that the option was specified in the correct case when it was issued.

In the message text:

option
The option specified on the rpcbind command.

System action

rpcbind stops.

Operator response

Contact the system programmer.

System programmer response

Restart rpcbind with a valid option. See the information about the RPCBIND in z/OS Communications Server: IP Configuration Reference for more information.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

rpcbind.c

Routing code

11

Descriptor code

0

Example

```
EZA4264I INCORRECT RPCBIND OPTION -X SPECIFIED
```

EZA4265I **RPCBIND *api* FAILED ERRNO = *errno* ERRNO2 = *errnojr***

Explanation

The rpcbind server attempted to use the API specified by C/C++. The interface returned -1 with *errno* and *errnojr*.

In the message text:

api
The failing C/C++ run-time library call name.

errno
The UNIX System Services return code. These return codes are listed in and described in the return codes (errnos) chapter in z/OS UNIX System Services Messages and Codes.

errnojr
The hexadecimal UNIX System Services reason code. The format of the 4-byte reason code is explained in the introduction to the reason codes (errnojrs) chapter of the z/OS UNIX System Services Messages and Codes, where the reason codes are listed.

System action

None.

Operator response

Contact the system programmer.

System programmer response

See z/OS XL C/C++ Runtime Library Reference to determine the reason for the returned value of the specified API. Determine and correct the API failure. Restart rpcbind.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

rpcbind.c

Routing code

Not applicable.

Descriptor code

Not applicable.

Example

```
RPCBIND BIND() FAILED ERRNO = 1115 ERRNO2 = 0X74637232
```

EZA4266I **LOST STATISTICS FOR PROGRAM = *program* VERSION = *version* PROC = *proc***

Explanation

The rpcbind server received a request to pass a call to another server on the rpcbind host.

The rpcbind server processed the request and tried to save statistics about the call. However, the rpcbind server was not able to save the statistics for that call.

If you use the rpcinfo utility from another host to query the rpcbind server statistics, the statistics reported will not include the program (*program*), version (*version*), or procedure (*proc*) information about this call. Likewise, if a program sends an RPCBPROC_GETSTAT request to the rpcbind server, the statistics reported will not include the program (*program*), version (*version*), or procedure (*proc*) information about this call.

In the message text:

program
The program number of the target program that was requested.

version
The version of the program that was requested.

proc
The procedure implemented by the program that was requested.

System action

Processing of the request completes. Statistics regarding the call are not saved.

Operator response

Save the system log for problem determination and contact the system programmer.

System programmer response

Obtain the system log and contact the IBM Support Center.

The rpcbind start option, -s, allows you to specify the number of statistics entries per binding protocol that the rpcbind server will maintain. Specifying a larger value for the -s option when starting rpcbind prevents the loss of statistics.

See the information about starting the rpcbind address space in z/OS Communications Server: IP Configuration Guide for more information about the -s parameter.

For more information about the RPCBPROC_GETSTAT request and about statistics regarding rpcbind calls to other servers, see RFC 1833. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

rpcbind.c

Routing code

1

Descriptor code

2

Example

```
EZA4266I LOST STATISTICS FOR PROGRAM= 200000 VERSION = 3 PROC=12
```

EZA4267I **PARAMETER *parameter* NOT VALID - VALUE MUST BE IN THE RANGE OF *low* - *high*.**

Explanation

You specified a parameter with a value that is not valid when you started the rpcbind server.

In the message text:

parameter
The name of the specified parameter.

low
The lowest value allowed for the specified parameter.

high
The highest value allowed for the specified parameter.

System action

The rpcbind server exits.

Operator response

Contact the system programmer.

System programmer response

See the information about starting the rpcbind address space in z/OS Communications Server: IP Configuration Guide for more information about the specified parameter. Ensure that the operator specifies a value within the acceptable range when starting the rpcbind server.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RPCBIND

Module

Not applicable.

Routing code

11

Descriptor code

0

Example

```
EZA4267I PARAMETER -s NOT VALID -  VALUE MUST BE IN THE RANGE OF 113 - 500.
```

EZA4268I **RPCBIND ENF SUPPORT FAILED RCODE *ret_code* REASON CODE *rsn_code***

Explanation

The rpcbind server could not establish Event Notification Facility (ENF) support. One of the following occurred:

- RPCBIND could not send an ENF system event indicating that it completed initialization.
- RPCBIND could not register a resource manager to send an ENF system event when RPCBIND ends.

RPC applications that use the ENF system event to determine when the rpcbind server is available might not register with this instance of the rpcbind server.

In the message text:

ret_code
Return code from ENF set up

rsn_code
Reason code from ENF set up

See ENF event code 80 return and reason codes in z/OS Communications Server: IP and SNA Codes for more information about *ret_code* and *rsn_code*.

System action

RPCBIND continues.

Operator response

Contact the system programmer.

System programmer response

See ENF event code 80 return and reason codes in z/OS Communications Server: IP and SNA Codes for more information about *ret_code* and *rsn_code*. If the problem persists, contact the IBM Support Center with the message.

User response

None.

Problem determination

See the system programmer response.

Source

z/OS Communications Server TCP/IP: rpcbind

Module

Rpcbind.c

Routing code

11

Descriptor code

0

Automation

This message goes to the system console. Automation can notify the system programmer to check on RPC applications that use the ENF system event to register with the rpcbind server.

Example

```
EZA4268I RPCBIND ENF NOTIFICATION FAILED RCODE 4 REASON CODE 00040008
```

EZA4275E **auth_unix.c - Fatal marshalling problem**

Explanation

The procedure authunix_create(), which creates and returns a handle that contains UNIX-based authentication information, was unsuccessful.

System action

authunix_create() call fails.

Operator response

No action is needed.

System programmer response

Check for the correct description in the authentication parameters of the authunix_create() and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

User response

Not applicable.

Problem determination

See System Programmer Response.

Source

z/OS Communications Server TCP/IP: RPC

Module

auth@uni.c

Routing code

11

Descriptor code

0

Example

Not applicable.

EZA4276E **get_myaddress: socket**

Explanation

The function get_myaddress(), which puts the local host's IP address into the *addr* parameter, was unsuccessful because of a negative return value received from the socket() system call. The socket() system call creates an endpoint for communication and returns a socket descriptor representing the endpoint.

System action

The calling program ends.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *domain*, *type*, and *protocol* parameters of the socket(&rparen call and the *addr* parameter of the get_myaddress() call, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@CLN

Procedure name

get_myaddress()

EZA4277E **get_myaddress: ioctl (get interface configuration)**

Explanation

The function get_myaddress(), which puts the local host's IP address into *addr*, was unsuccessful because of a negative return value received from the ioctl() system call. The ioctl() system call controls the operating characteristics of sockets.

System action

The calling program ends.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *s*, *cmd*, and *arg* parameters of the ioctl() system call and the *addr* parameter of the get_myaddress() call, check that the network interface configuration is defined in an *ifconf* structure of the IF.H data set, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@CLN

Procedure name

get_myaddress()

EZA4278E **get_myaddress: ioctl**

Explanation

The function get_myaddress(), which puts the local host's IP address into *addr*, was unsuccessful because of a negative return value received from the ioctl() system call. The ioctl() system call controls the operating characteristics of sockets.

System action

The calling program ends.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *s*, *cmd*, and *arg* parameters of the ioctl() system call and the *addr* parameter of the get_myaddress() call, check that the network interface flags are defined in an *ifreq* structure of the IF.H data set, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@CLN

Procedure name

get_myaddress()

EZA4279E **broadcast: ioctl (get interface configuration)**

Explanation

The procedure getbroadcastnets(), which gets the address of a network to broadcast a remote procedure, was unsuccessful because of a negative return code received from the ioctl() system call. The ioctl() call controls the operating characteristics of sockets.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *s*, *cmd*, and *arg* parameters of the ioctl() system call and the parameters of the clnt_broadcast() call, check that the network interface configuration is defined in an *ifconf* structure of the IF.H data set, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

getbroadcastnets()

EZA4280E **broadcast: ioctl (get interface destination address)**

Explanation

The procedure getbroadcastnets(), which gets the address of a network to broadcast a remote procedure, was unsuccessful because of a negative return code received from the ioctl() system call. The ioctl() call controls the operating characteristics of sockets.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *s*, *cmd*, and *arg* parameters of the ioctl() system call and the parameters of the clnt_broadcast() call, check that the network interface destination address is defined in an *ifreq* structure of the IF.H data set, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

getbroadcastnets()

EZA4281E **broadcast: ioctl (get interface broadcast address)**

Explanation

The procedure getbroadcastnets(), which gets the address of a network to broadcast a remote procedure, was unsuccessful because of a negative return code received from the ioctl() system call. The ioctl() call controls the operating characteristics of sockets.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *s*, *cmd*, and *arg* parameters of the ioctl() system call and the parameters of the clnt_broadcast() call, check that the network interface broadcast address is defined in an *ifreq* structure of the IF.H data set, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

getbroadcastnets()

EZA4282E **Cannot create socket for broadcast rpc**

Explanation

The procedure clnt_broadcast(), which broadcasts the remote procedure described by *prognum*, *versnum*, and *procnum* to all locally connected broadcast network, was unsuccessful because of a negative return code received from the socket() system call. The socket() call creates an endpoint for communication and returns a socket descriptor representing the endpoint.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the clnt_broadcast() and the socket() parameters and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

clnt_broadcast()

EZA4283E **Cannot send broadcast packet**

Explanation

The procedure clnt_broadcast was unsuccessful because of a negative return code received from the sendto() system call. Destruction usually involves freeing private data structures associated with the stream. The sendto() system call sends datagrams on the socket with the descriptor *s*. This call applies to any datagram socket, whether connected or unconnected.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the xdr_destroy() and sendto() calls and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

clnt_broadcast

EZA4284E **Broadcast select problem**

Explanation

The broadcast of a remote procedure to all locally connected broadcast network was unsuccessful because of a negative return code received from the select() system call. The select() call monitors activity on a set of sockets to see if any of the sockets are ready for reading, writing, or have an exceptional condition pending.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the select(), xdr_destroy(), and clnt_broadcast() calls and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

xdr_destroy()

EZA4285E **Cannot receive reply to broadcast**

Explanation

The broadcast of a remote procedure to all locally connected broadcast network was unsuccessful because of a negative return code received from the recvfrom() system call. The recvfrom() call receives data on a socket with descriptor *s* and stores it in a buffer. This call applies to any datagram socket, whether connected or unconnected.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the recvfrom(), xdr_destroy(), and clnt_broadcast() calls and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@RMT

Procedure name

clnt_broadcast

EZA4286E **svc_run: - select failed**

Explanation

The function svc_run(), which accepts RPC requests and calls the appropriate service, was unsuccessful because of a negative return code received from select() system call. The select() call monitors activity on a set of sockets to see if any of the sockets are ready for reading, writing, or have an exceptional condition pending.

System action

Svc_run() returns to the caller.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *nfds*, *readfds*, *writefds*, *exceptfds*, and *timeout* parameters of the select() system call and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@RUN

Procedure name

svc_run()

EZA4287E **svctcp_.c - tcp socket creation problem**

Explanation

The function svctcp_create(), which creates a TCP-based service transport, was unsuccessful because of a negative return code received from the socket() system call. The socket() call creates an endpoint for communication and returns a socket descriptor representing the endpoint. When RPC_ANYSOCK is specified in the *sock* parameter of the svctcp_create() call, a new socket is created. If the socket is not bound to a local TCP port, it is bound to an arbitrary port.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the svctcp_create() and socket() calls, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@TCP

Procedure name

svctcp_create()

EZA4288E **svctcp_.c - cannot getsocketname or listen**

Explanation

The function svctcp_create(), which creates a TCP-based service transport, was unsuccessful because of a negative return code received from the getsockname() or the listen() system call. The getsockname() call stores the current name for the socket specified by the *s* parameter into the structure pointed to by the *name* parameter. It returns the address of the socket that has been bound. The listen() call applies only to stream sockets. This call completes the binding necessary for a socket and it creates a connection request queue to queue incoming connection requests.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the svctcp_create(), getsockname(), and listen() calls, and restart the program. See z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@TCP

Procedure name

svctcp_create()

EZA4289E **svcudp_create: socket creation problem**

Explanation

The function svcudp_create(), which creates a UDP-based service transport, was unsuccessful because of a negative return code received from the socket() system call. The socket() call creates an endpoint for communication and returns a socket descriptor representing the endpoint. When RPC_ANYSOCK is specified in the *sock* parameter of the svcudp_create() call, a new socket is created.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the svcudp_create() and socket() calls, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@UDP

Procedure name

svcudp_bufcreate()

EZA4290E **svcudp_create - cannot getsocketname**

Explanation

The function svcudp_create(), which creates a UDP-based service transport, was unsuccessful because of a negative return code received from the getsockname(). The getsockname() call stores the current name for the socket specified by the *s* parameter into the structure pointed to by the *name* parameter. It returns the address of the socket that has been bound.

System action

An error is returned to the calling program.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the parameters of the svcudp_create(), getsockname(), and listen() calls, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@UDP

Procedure name

svcudp_bufcreate()

EZA4291E **broadcast: ioctl (get interface flags)**

Explanation

The procedure getbroadcastnets(), which gets the address of a network to broadcast a remote procedure, was unsuccessful because of a negative return code received from the ioctl() system call. The ioctl() call controls the operating characteristics of sockets.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check for the correct entries declared in the *s*, *cmd*, and *arg* parameters of the ioctl() system call and the parameters of the clnt_broadcast() calls, check that the network interface flags are defined in an *ifreq* structure of the IF.H data set, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

PMAP@CLN

Procedure name

getbroadcastnets()

EZA4301E ***** Can't initialize resolver.**

Explanation

The Domain Name Server was unable to initialize the resolver library routines, which are used by clients to request resolution by the Domain Name Server. This indicates that there was an error in specifying the configuration file for the resolver, or the Domain Name Server was unable to allocate the configuration data set. The default name for the configuration data set is *hlq*.RESOLV.CONF.

System action

Initialization stops. TCPIP continues.

Operator response

Check the syntax of the configuration data set and resubmit the command. If the error persists, specify the configuration options using the NSLOOKUP command and its parameters. For more information about the configuration data set, see z/OS Communications Server: IP Configuration Reference. For more information about the NSLOOKUP command and its parameters, see the z/OS Communications Server: IP System Administrator's Commands.

System programmer response

Make more storage available to the Domain Name Server if necessary.

EZA4302I **Usage: *command* infile *command***

Explanation

The command line syntax was incorrect; the correct command line syntax is displayed. *Command* is replaced by the name of this program, RPCGEN, unless the executable’s file name was changed.

System action

RPCGEN exits.

Operator response

Correct the syntax and restart.

System programmer response

No action is needed.

Module

RPCMAIN

Procedure name

Main()

EZA4303I **[-c | -h | -l | -m] [-o fn ft fm] [in fn in ft [in fm]]**

Explanation

The command line syntax was incorrect. This message indicates the correct syntax for this command.

System action

RPCGEN continues.

Operator response

Enter the command using the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

RPCMAIN

Procedure name

main()

EZA4304I ***command* [-c | -h | -l | -m] [-o *outfile*] [*infile*]**

Explanation

The command line syntax was incorrect. This message indicates the correct syntax for this command.

System action

RPCGEN continues.

Operator response

Enter the command using the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

RPCMAIN

Procedure name

main()

EZA4306I ***command* [-s udp | tcp] * [-o *outfile*] [*infile*]**

Explanation

The command line syntax was incorrect. This message indicates the correct syntax for this command.

System action

RPCGEN continues.

Operator response

Enter the command using the correct syntax and try it again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

RPCMAIN

Procedure name

main()

EZA4307E ***command*: output would overwrite *file***

Explanation

The specified output file would overwrite the input file. The output file name should be different from the input file name.

System action

RPCGEN exits.

Operator response

Change the output file name and restart.

System programmer response

No action is needed.

Module

RPCMAIN

Procedure name

Open_output()

EZA4308E ***command*: unable to open *file rc***

Explanation

RPCGEN was unable to open the output file.

System action

RPCGEN exits.

Operator response

Determine from the error code why the file is unavailable. Possible reasons include: file missing, out of storage, and wrong pointer.

System programmer response

No action is needed.

Module

RPCMAIN

Procedure name

Open_output()

EZA4309E ***line file*, line *number*: preprocessor error**

Explanation

RPCGEN encountered a preprocessor error while searching for the null at the end of the number and spaces. A non-null character was found at the end of the number and spaces.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

Correct the program.

Module

RPCSCAN

Procedure name

Docppline()

EZA4310E ***line file*, line *number*: constant or identifier expected**

Explanation

RPCGEN expected a constant or identifier at this point in the source file.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

No action is needed.

Module

RPCSCAN

Procedure name

Scan_num()

EZA4311E ***line file*, line *number*: illegal character in file**

Explanation

RPCGEN encountered an incorrect character in the source file.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

Correct the program.

Module

RPCSCAN

Procedure name

Get_token()

EZA4312E ***line file*, line *number*: unterminated string constant**

Explanation

RPCGEN encountered an unterminated string constant. The string should be terminated with a null character (\0 or X'00'). The program could be overwriting data.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

Examine the program data for an unterminated string or an incorrect pointer. Correct the program.

Module

RPCSCAN

Procedure name

Findstrconst()

EZA4313E ***line file*, line *number*: preprocessor error**

Explanation

RPCGEN encountered an error during preprocessing. This message indicates the data set name and the line number where the error occurred.

System action

RPCGE continues.

Operator response

No action is needed.

System programmer response

Correct the line number, specified in this message, in the source data set and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

RPCSCAN

Procedure name

docppline()

EZA4314E **Too many files!**

Explanation

The number of open files exceeds the maximum permitted. The maximum number of files allowed is 4. The program depends upon the C compiler to clear the variable *nfiles* to 0. If C does not clear *nfiles* to 0, the message occurs unpredictably.

System action

RPCGEN exits.

Operator response

No action is needed.

System programmer response

If the compiler does not provide a reset to *nfiles*, add a software reset to *nfiles*.

Module

RPCUTIL, RPCMAIN

Procedure name

Record_open(), main()

EZA4318E ***line file*, line *number*: definition keyword expected**

Explanation

RPCGEN expected a definition keyword at this point in the source file.

System action

RPCGEN exits.

Operator response

Correct the source file and restart.

System programmer response

No action is needed.

Module

RPCPARSE

Procedure name

Get_definition()

EZA4319E ***line file*, line *number*: illegal result type**

Explanation

RPCGEN encountered an incorrect result type, Opaque, on this line of the source file.

System action

RPCGEN exits.

Operator response

Correct the source file and restart.

System programmer response

No action is needed.

Module

RPCPARSE

Procedure name

Def_program()

EZA4320E ***line file*, line *number*: illegal argument type**

Explanation

RPCGEN encountered an incorrect argument type, Opaque, on this line of the source file.

System action

RPCGEN exits.

Operator response

Correct the source file and restart.

System programmer response

No action is needed.

Module

RPCPARSE

Procedure name

Def_program()

EZA4321E ***line file*, line *number*: no array-of-pointer declarations — use typedef**

Explanation

RPCGEN encountered an incorrect array-of-pointer declaration at this line in the source file. RPCGEN does not allow arrays of pointers, but they can be implemented by means of typedef.

System action

RPCGEN exits.

Operator response

Do a typedef of the array-of-pointers, or tell the system programmer about the error.

System programmer response

Do a typedef of the array-of-pointers and restart.

Module

RPCPARSE

Procedure name

Get_declaration()

EZA4322E ***line file*, line *number*: array declaration expected**

Explanation

RPCGEN expected an array declaration at this line of the source file.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

Correct the program.

Module

RPCPARSE

Procedure name

Get_declaration()

EZA4323E ***line file*, line *number*: variable-length array declaration expected**

Explanation

RPCGEN expected a variable-length array declaration at this point in the source file.

System action

RPCGEN exits.

Operator response

Correct the input program and restart; or tell the system programmer about the error.

System programmer response

Correct the input program.

Module

RPCPARSE

Procedure name

Get_declaration()

EZA4324E ***line file*, line *number*: voids allowed only inside union and program definitions**

Explanation

RPCGEN encountered an inappropriate void type at this line in the source file.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

Correct the program.

Module

RPCPARSE

Procedure name

Get_type()

EZA4325E ***line file*, line *number*: expected type specifier**

Explanation

RPCGEN expected a type specifier at this point in the source file.

System action

RPCGEN exits.

Operator response

Correct the program and restart; or tell the system programmer about the error.

System programmer response

Correct the program.

Module

RPCPARSE

Procedure name

Get_type()

EZA4326I **Program *program* is not available**

Explanation

The program name specified in the *prognum* parameter of the clntudp_create() procedure is not available. This procedure call creates a client transport handle for the remote program (*prognum*). UDP is used as the transport.

System action

The system continues.

Operator response

No action is needed.

System programmer response

Specify the correct program name in the *prognum* parameter of the clntudp_create() procedure call, check that the program name specified is available in storage and resubmit the call. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

RPCINFO

Procedure name

udpping()

EZA4327I **Program *number* version *version* is not available**

Explanation

The indicated program is not ready.

System action

The system continues.

Operator response

Start the server that uses the required program number.

System programmer response

No action is needed.

Module

RCPINFO

Procedure name

Pmapdump()

EZA4328I **Program *number* version *version* ready and waiting**

Explanation

The indicated program is ready.

System action

The system continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RPCINFO

Procedure name

Pmapdump()

EZA4329X **Rpcinfo: can't contact portmapper (EZA4329)**

Explanation

RPCINFO encountered an error establishing a TCP connection to Portmapper. See z/OS Communications Server: IP and SNA Codes for more information on System Return Codes.

System action

RPCINFO exits.

Operator response

Correct the problem identified in the error response. A typical error is that the Portmapper is not running on the host. Start PORTMAP. PING the host to determine that Portmapper is running and available on the network.

System programmer response

No action is needed.

Module

RPCINFO

Procedure name

Pmapdump(), clnt_call() brdcst(), clnt_broadcast()

EZA4330I **No remote programs registered.**

Explanation

No remote programs have been registered on the Portmapper to be retrieved by RPCINFO.

System action

RPCINFO continues.

Operator response

Start a server that registers a procedure on the Portmapper.

System programmer response

No action is needed.

Module

RPCINFO

Procedure name

Pmapdump()

EZA4331E **Usage: rpcinfo -n portnum u host prognum versnum**

EZA4332I **rpcinfo -n portnum t host prognum versnum**

EZA4333I **rpcinfo p host**

EZA4334I **rpcinfo -b prognum versnum**

Explanation

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

System action

RPCINFO exits.

Operator response

Reenter RPCINFO with the correct control parameter syntax.

System programmer response

No action is needed.

Module

RPCINFO

Procedure name

Usage()

EZA4335E **Rpcinfo: *service* is unknown service**

Explanation

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

System action

RPCINFO exits.

Operator response

Correct the program number. Check that the server is running and that it registers the correct number. Rerun.

System programmer response

No action is needed.

Module

rpcinfo

Procedure name

Getprognum(), getrpcbyname()

EZA4336E **Rpcinfo: *host* is unknown host**

Explanation

You specified an unrecognized host on the RPCINFO command line. The name server might be down. The host name might have been changed to another network. The *hlq*.HOSTS.LOCAL, *hlq*.HOSTS.ADDRINFO, or *hlq*.HOSTS.SITEINFO data sets might be incorrect. You might need to run MAKESITE.

System action

RPCINFO exits.

Operator response

PING the host. Determine whether the host name is correct and whether the name server is running.

System programmer response

Make any necessary changes to the *hlq*.HOSTS.LOCAL data set and run MAKESITE.

Module

RPCINFO

Procedure name

Get_net_address() udpping(), socket() udpping(), tcpping()

EZA4337E **Rpcinfo: broadcast failed: *rc***

Explanation

RPCINFO encountered an error issuing a broadcast.

System action

RPCINFO exits.

Operator response

Use *err* in the message to determine the probable cause. There are many possible return codes. See clnt_stat in z/OS Communications Server: IP Programmer's Guide and Reference.

System programmer response

No action is needed.

Module

RPCINFO

Procedure name

Brdcst(), clnt_broadcast()

EZA4338E ***procedure:* out of memory**

Explanation

The procedure specified was unsuccessful because of insufficient storage allocated for completion.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Increase the region size of the application executing the remote procedure call (RPC).

Module

AUTH@UNI, CLNT@TCP, CLNT@UDP, SVC@SIMP, SVC@TCP, SVC@UDP, XDR, XDR@ARRA, XDR@REC, XDR@REFE, XDR@STR

Procedure name

authunix_create(), clnttcp_create(), mem_alloc(), clntudp_bufcreate(), registerrpc(), malloc(), makefd_xprt(), svctcp_create(), svcudp_butcreate(), xdr_array(), xdr_bytes(), xdr_reference(), xdr_string(), xdrrec_create()

EZA4339E ***message***

Explanation

The procedure clnt_sperror(), which returns the address of a message indicating why an RPC did not complete, returned the displayed message. This procedure receives pointers previously declared in the *clnt* parameter of the clnt_create(), clntudp_create(), clnttcp_create(), or clntraw_create() procedures. The *clnt* parameter is the pointer to a client handle that was obtained using one of the above procedures. Depending on the location where the procedure stopped, only one of these messages will be displayed.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Possible responses for this message are:

Message
: **Response**

RPC: Success
: None.

RPC: Can't encode arguments
: Check for the correct XDR procedure name in the *inproc* parameter of the callrpc() call and restart the program.

RPC: Can't decode result
: Check for the correct XDR procedure name in the *outproc* procedure of the callrpc() call and restart the program.

RPC: Unable to send
: Check for the correct parameter entries in the callrpc() call and restart the program.

RPC: Unable to receive; errno = *value*, *error list*
: Check for the correct parameter entries in the callrpc() call and restart the program.

RPC: Timed out
: Restart the program if required.

RPC: Incompatible versions of RPC; low version = *version*, high version = *version*
: Check for the correct remote program version number in the *versnum* parameter of the callrpc() call and restart the program.

RPC: Authentication error; why = *reason*
Check for the correct entries in the authentication parameters of the authunix_create() call.

RPC: Authentication error; why = unknown authentication error - *value*
Check for the correct entries in the authentication parameters of the authunix_create() call.

RPC: Program unavailable
Check for the correct program number in the *prognum* parameter of the callrpc() call, register the program using the registerrpc() call, or contact the remote system administrator to obtain access to the program and restart the program.

RPC: Program/version mismatch; low version = *version*, high version = *version*, *version*, *version*
Check for the correct remote program number and version in the *prognum* and *versnum* parameters of the callrpc() and restart the program.

RPC: Procedure unavailable
Check for the correct procedure number in the *procnum* parameter of the callrpc() call and restart the program.

RPC: Server can't decode arguments
Check for the correct XDR routine in the *inproc* parameter of the svc_freeargs() call and restart the program.

RPC: Remote system error *errcode*
Use the system error code displayed in this message and the list of system error codes in z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference to determine the cause of the error and inform the remote system administrator.

RPC: Unknown host
Check for the correct host ID in the *host* parameter of the callrpc() call and restart the program.

RPC: Unknown protocol
Check for the transport protocol in the *-s transport* parameter of the RPCGEN command and verify that both TCP and UDP are specified in the /etc/protocol directory. TCP and UDP are the supported transports. A sample copy of the /etc/protocol directory is shipped in the /usr/lpp/tcpip/samples/protocol directory.

RPC: Port mapper failure
Check that the *tcpip.*SEZAINST(ETCRPC) data set contains an entry for the portmapper procedure name and restart the program.

RPC: Program not registered
Check for the correct program number in the *prognum* parameter of the svc_register() call. This parameter specifies the program number to be registered.

RPC: Failed (unspecified error)
Contact the IBM Software Support Center.

RPC: default; sl=*value*, s2=*value*, *message*, *message*
Contact the IBM Software Support Center.

See Appendix A, “Additional information about remote procedure call (RPC) messages,” on page 1063 for more information about tcperror return codes. For more information about the RPC and XDR protocols, see the Sun Microsystems publication, *Networking on the Sun Workstation: Remote Procedure Call Programming Guide*.

Module

CLNT@PER

Procedure name

clnt_sperror

EZA4340E **clnt_raw.c - Fatal header serialization error.**

Explanation

The procedure call clntraw_create, which creates a dummy RPC client for the remote program , was unsuccessful because a call issued to xdrmem_create() returned an incorrect return code. The xdrmem_create() call initializes an XDR stream in local storage.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Check for the correct remote program number and version in the *prognum* and *versnum* parameters of the clntraw_create() call and the parameters specified in the xdrmem_create() call, and restart the program.

Module

CLNT@RAW

Procedure name

clntraw_create

EZA4341I **clnt_tcp: readtcp: read returns=*value***

Explanation

The procedure clnttcp, which creates a remote procedure call (RPC) client transport handle for the remote program specified in the *prognum, versnum* parameters, has received the specified return value from the readtcp procedure. The readtcp() reads data contained in the TCP data buffer. This procedure is an interface between the external data representation (xdr) serializer and the TCP connection. TCP is used as the transport layer.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CLNT@TCP

Procedure name

readtcp

EZA4342I **clnt_tcp: attempting to write *number* bytes**

EZA4343I ***call name*: *function*: buffer dump**

Explanation

The procedure clnttcp, which creates a remote procedure call (RPC) client transport handle for the remote program specified in the *prognum, versnum* parameters, is trying to write the specified number of bytes received from TCP. These parameters specify the remote program number and the version number of the remote program. This procedure is an interface between the external data representation (XDR) serializer and the TCP connection. TCP is used as the transport layer.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CLNT@TCP

Procedure name

readtcp

EZA4344E **Can't reassign procedure number *procedure***

Explanation

The registerrpc() call registers a procedure with the local portmapper, and creates a control structure to remember the server procedure and its external data representation (XDR) routine. Because of an incorrect procedure number specified in the *procnum* parameter, this call did not complete.

System action

TCPIP continues.

Operator response

Inform the system program about this message.

System programmer response

Check for the correct procedure number and restart the program. Procedures registered using registerrpc() are accessed using the UDP transport layer. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@SIMP

Procedure name

registerrpc()

EZA4345E **Couldn't create an rpc server**

Explanation

A request to create an RPC server was unsuccessful because of an incorrect pointer declared in the *sockp* parameter of the svcudp_create() call. This parameter specifies the pointer to the socket associated with the service transport handle. If *sockp* is RPC_ANYSOCK, a new socket is created. This procedure creates a UDP-based service transport to which it returns a pointer. The UDP transport layer is used to access the procedures registered in the registerrpc() call.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Specify the correct pointer in the *sockp* parameter of the svcudp_create() call. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@SIMP

Procedure name

registerrpc()

EZA4346E **Couldn’t register prog *number* vers *number***

Explanation

A request to have RPC register this program and version was unsuccessful because portmapper might not be running or sufficient storage might not be available.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Use PING and RPCINFO to determine whether portmapper is running, allocate more storage if necessary, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@SIMP

Procedure name

pmap_unset

EZA4347E **svc_sendreply failed**

Explanation

The svc_sendreply(), which is called by the service dispatch routine to send the results of the call to the caller, was unsuccessful.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Remove unneeded tasks to free storage, check for an incorrect pointer, or check that storage is sufficient, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@SIMP

Procedure name

universal()

EZA4348E **Trouble replying to prog *number***

Explanation

The procedure svc_sendreply, which is called by the service dispatch routine to send the results of the call to the caller, was unsuccessful because of an incorrect pointer declared in the *xprt* parameter or not enough storage was allocated for the procedure to complete. The *xprt* parameter is the pointer to the caller's transport handle.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Specify the correct pointer in the *xprt* parameter of the svc_ sendreply procedure, check whether the task has adequate storage, and restart the program. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@SIMP

Procedure name

universal()

EZA4349E **Never registered prog *number***

Explanation

An attempt was made to communicate with an unregistered program.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Use PING and RPCINFO to determine whether the host is up and the program is registered. If the program is not registered, register it on the target host. See the z/OS Communications Server: IP Programmer's Guide and Reference for more information.

Module

SVC@SIMP

Procedure name

universal()

EZA4350E **Bad auth_len gid *length* str *length* auth *length***

Explanation

An RPC server received authorization credentials of an incorrect length. Authorization is denied. This message indicates the length specified in the *len* parameter of the authunix_create() call. This parameter is the length of the information pointed to by *aup_gids*.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Check for the correct authorization credentials entries in the authunix_create() call and restart the program.

Module

SVC@AUUN

Procedure name

Svcauth_unix()

EZA4351I **xdr_u_long decode gets *value*, returns *value***

Explanation

The procedure xdr_u_long(), which translates between C unsigned long integers and their external representations, has been initiated. This message indicates the value that has been decoded.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

XDR

Procedure name

xdr_u_long

EZA4352I **xdr_u_long encode adds *value*, returns *value***

Explanation

The procedure xdr_u_long(), which translates between C unsigned long integers and their external representations, has been initiated. This message indicates the value that has been encoded.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

XDR

Procedure name

xdr_u_long

EZA4353I **Alarm called with argument: *value*; old value = *value*.**

Explanation

A timer has been set by the svc_run function. This message indicates the value supplied to this function for use in a select() system call. When the *timeout* parameter of the select() call is not a NULL pointer, it specifies a maximum interval to wait for the selection to complete.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SVC@RUN

Procedure name

svc_run()

EZA4354I **svc_run: call select with timeout = *time***

Explanation

The timeout value points to the time to wait for the select() system call to complete. When the *timeout* parameter is not a NULL pointer, it specifies a maximum interval to wait for the selection to complete. The select call then returns either when a new client request arrives, or when the *time* value ends, whichever occurs first.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SVC@RUN

Procedure name

svc_run

EZA4355I **svc_run: calling select with no timeout.**

Explanation

No timeout has been set by the svc_run for the select call. When the *timeout* parameter is a NULL pointer, the select call blocks until a socket becomes ready.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SVC@RUN

Procedure name

svc_run()

EZA4356I **svc_run: select returns value *time* / *NULL***

Explanation

Indicates the value of the *timeout* parameter returned by the select() system call. The select() call monitors activity on a set of sockets to see if any of the sockets are ready for reading, writing, or have an exceptional condition pending. Depending on the value specified only one of these variables will be displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SVC@RUN

Procedure name

svc_run()

EZA4359I **sigsetmask: new mask = *value*, old mask = *value***

Explanation

Indicates the value of the old and new masks that have been set by the sigsetmask() function. This function is called when the pointer specified in the *timeout* parameter of the select() system call expires.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSIGNAL

Procedure name

sigsetmask

EZA4360I **sigblock: arg = *value*, new mask = *value*, old mask = *value*.**

Explanation

The function sigblock, which adds the specified signals to the current mask, has been initiated with the displayed values.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSIGNAL

Procedure name

sigblock

EZA4361I **rraise: arg = *value*.**

Explanation

The rraise function, which invokes the specified signal handler synchronously, has been initiated. This function is called when the time specified in the *timeout* parameter of the select() system call returns a value of 0 indicating that a timeout has occurred.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSIGNAL

Procedure name

rraise

EZA4375E **rexecd: fread: *rc***

Explanation

An error was detected while the spool data set was being read. This message indicates the return code received for this procedure.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check the REXEC command format when submitting REXEC as a batch job. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

MVRSHD

Procedure name

main

EZA4376E **rshd: remuser too long**

Explanation

The user ID specified in the *-l user/password* parameter of the RSH command is too long. The connection is closed. The maximum length for this parameter is 15 characters. This parameter specifies the user ID on the remote host.

System action

Processing continues.

Operator response

Check for the correct user ID in the *-l user/password* parameter of the RSH command and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4377E **rexecd: username too long**

Explanation

The user ID specified in the *-l user_id* parameter of the REXEC command is too long. The connection is closed. The maximum length for this parameter is 15 characters. This parameter specifies the user ID on the remote host.

System action

Processing continues.

Operator response

Check for the correct user ID in the *-l user_id* parameter of the REXEC command and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4378E **rshd: username too long**

Explanation

The user ID specified in the *-l user_id* parameter of the REXEC command is too long. The connection is closed. The maximum length for this parameter is 15 characters. This parameter specifies the user ID on the local host.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check for the correct user ID in the *-l user_id* parameter of the REXEC command and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

MVRSHD

Procedure name

main

EZA4379E **rexecd: password too long**

Explanation

The password declared in the *-p password* parameter of the REXEC command or the *-l user/password* parameter of the RSH command is too long. The connection is closed. This parameter specifies the password for the user ID on the remote host. The maximum length is 15 characters.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check for the correct password in the *-p password* parameter of the REXEC command or the *-l user/password* parameter of the RSH command and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

MVRSHD

Procedure name

main

EZA4380E **rshd: Host unknown.**

Explanation

The IP address of the machine issuing the RSH command could not be resolved to a hostname. getnameinfo() failed.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Ensure that the IP address and host name of remote clients can be determined by resolver services. Restart the connection if required, and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

MVRSHD

Procedure name

doit

EZA4381I **Accept *socket_number* from *remote_address* on *local_address***

Explanation

The server has accepted a connection on the specified socket from the remote IP address to the local IP address.

In the message text:

socket_number
The number of the socket from which the connection was being made.

remote_address
The remote IP address and port number.

local_address
The local IP address and port number.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

No action is needed.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RSHD

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA4381I Accept 2 from 9.44.36.163:1023 on 9.42.105.158:514
```

EZA4382I **Connecting on *socket number* to *remote_address* from *local_address***

Explanation

The server is opening a connection on the specified socket to the remote IP address from the local IP address.

In the message text:

socket_number
The number of the socket from which the connection was being made.

remote_address
The remote IP address and port number.

local_address
The local IP address and port number.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RSHD

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable for automation.

Example

```
EZA4382I Connecting on 4 to 9.44.36.163:3291 from 9.42.105.158:1037
```

EZA4383I **SSCSARAY *jobid flag***

Explanation

The progress of the job is being tracked. This message indicates the job name and the value of the flag. The 2-digit number following the job name is a byte with the following bit pattern:

```
1         Job is currently active
 1        Job is waiting for processing
  1       Job is on the output queue
   1      Job is held in its current queue
     1    Job has a second-level message
      1   Job active in NJE
       11 Unused
```

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4384I **SSCSRT: *rc jobname job ID***

Explanation

A cancel request has been received. This message displays the return code, the job name, and the job ID of the job currently running or waiting to be run. The job name and ID are the same as the job to be submitted.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4385I **SSSORT(CTRL):** ***rc***

Explanation

Displays the return code from an SSSO request with the CTRL bit on. This request is used to tell JES that RSHD is starting to look for output from a job. SSSO is the subsystem option block extension for processing SYSOUT data sets.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4386E **rshd: Permission denied.**

Explanation

This message is displayed for one of the following reasons:

- The RSH command issued from the client did not include a valid password.
- The *userid.*RHOSTS.DATA data set does not exist or was inaccessible to the REXEC Daemon.
- The *userid.*RHOSTS.DATA data set does not contain the host name and user ID for the client host where the command was issued.

A message is sent to the RSH connection and the connection is closed.

System action

Processing continues.

Operator response

Issue the RSH command with a valid password, or make sure that the *userid.*RHOSTS.DATA data set exists and contains both the host name and user ID of the client system.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

Not applicable.

EZA4387I **SSSORT(del): *rc***

Explanation

Displays any nonzero return codes received when output from a previous job, which might be interpreted as output for this job, is found. These files are deleted when found.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4388I **SSSORT(fail): *rc***

Explanation

The output data set could not be opened. The job that was submitted is purged. This message indicates the return code received after this procedure.

System action

Processing continues.

Operator response

Check for the correct parameters declared in the output data set and submit the job again. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4389I **SSSORT(init):** ***rc***

Explanation

Displays the return code received after looking for the first output to be directed back to the terminal. If there is no output, either this is an old job with the same name, or the output for the correct job is not available yet.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4390I **SSSORT(next):** ***rc***

Explanation

Checking to see if there is any more work to be done. This is usually the last message you will see and indicates that the return code was nonzero. The various possibilities for the return code are:

0
Successful completion - request went to the subsystem.

4
Subsystem does not support this function. This is the one that is expected when there is no more work to do.

8
Subsystem exists, but is not up.

12
Subsystem does not exist.

16
Function not completed - irrecoverable error.

20
Logical error (for example, incorrect SSOB format, length.)

System action

Processing continues.

Operator response

If the return code is other than 0 or 4, contact the system programmer.

System programmer response

If the return code is other than 0 or 4, determine the cause of the error using the return codes above. If the error persists, contact the IBM Software Support Center.

Module

MVRSHD

Procedure name

main

EZA4391I **SSSORT(tso):** ***rc***

Explanation

Indicates the return code received after a TSO command has been encountered.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4392I **S99ret:** ***rc destination-class job-name data-set-name***

Explanation

The return code from an SVC 99 where the data set is being allocated is listed first, followed by the destination class, the job name, and the SYSOUT data set name in that order.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4393I **S99ret: *rc***

Explanation

This message posts the return code for the SVC 99 deallocation of the data set that was named in message EZA4392I.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4394I **Writing record *record*, *bytes* bytes.**

Explanation

While attempting to write data to the socket an error occurred. This is most likely an internal processing error.

System action

Processing continues.

Operator response

Inform the system programmer of the error.

System programmer response

If the error can be repeated, contact the IBM Software Support Center with the console log from the server.

Module

MVRSHD

Procedure name

main

EZA4395E **Syntax error: *error***

Explanation

A command has been entered that contains a syntax error.

System action

Processing continues with the previously defined options.

Operator response

Enter the command with the correct syntax and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4396E **MAXCONN parameter may not be set after START.**

Explanation

A MAXCONN parameter was specified with a MODIFY command after the server had already been started. MAXCONN specifies the maximum number of open sockets at any one time. Usually, each client requires 2 sockets while the command is being processed and the output is being returned. The default is 512. MAXCONN must be specified when the server is started.

System action

Processing continues with the value declared for the MAXCONN parameter at the time of initialization.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4397E **Insufficient storage**

Explanation

There is not enough storage for the allocation of the fd_table to complete.

System action

The program ends. Processing continues.

Operator response

No action is needed.

System programmer response

Increase the region size of the application or lower the number of open sockets in the MAXCONN= parameter. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

MVRSHD

Procedure name

main

EZA4398E **Host unknown: *client name*; tracing ALLCLIENTS**

Explanation

The TRACE=*client name* parameter has been specified for the indicated client. However, the client name cannot be resolved by the name server or found in the HOSTS.SITEINFO data set.

System action

Tracing is set to ALLCLIENTS. Processing continues.

Operator response

Check for the correct client name and resubmit the command. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

Process_options

EZA4399E **Invalid trace option *value***

Explanation

An incorrect option has been specified for the TRACE= parameter. The following list provides the valid options for this parameter:

LOG/NOLOG
SEND/NOSEND
CLIENT=*client*
ALLCLIENTS
RESET

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Specify the correct option for the TRACE= parameter. If more than one option is specified, enclose the options in parentheses. These trace options can be changed with a MODIFY command. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

MVRSHD

Procedure name

process_options

EZA4400I **Trace options: *value*LOG,*value*SEND,ALLCLIENTS**

Explanation

Indicates the options that have been specified for the TRACE= parameter and that are in effect for the REXECD server.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4401I **Trace options: *value*LOG,*value*SEND, CLIENT=*client***

Explanation

Indicates the options that have been specified for the TRACE= parameter and that are in effect for the REXECD server.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4402E **Invalid option *option***

Explanation

An incorrect option has been specified for the TRACE= parameter. The following list provides the valid options for this parameter:

LOG/NOLOG
SEND/NOSEND
CLIENT=*client*
ALLCLIENTS
RESET

The default values for the TRACE= parameter are used.

System action

Processing continues.

Operator response

Specify the correct option for the TRACE= parameter. If more than one option is specified, enclose the options in parentheses. These trace options can be changed with a MODIFY command. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4403I **Parameters: MSGCLASS=*class*,TSCLASS=*class*, TSOPROC=*value*,MAXCONN=*value*, EXIT=*value***

Explanation

Indicates the values that have been specified for the parameters of the RXSERVE cataloged procedure. The following list provides a description of these parameters:

EXIT=
Name of user exit routine to inspect and alter JOB and EXEC parameters prior to submission of TSO batch jobs initiated by remote commands.

TSOPROC=
The name of the TSO batch procedure. The default is IKJACCNT. The name IKJACCNT can be modified in the exit routine specified with the EXIT parameter.

MSGCLASS=
The MSGCLASS parameter for TSO batch jobs submitted to execute remote commands. The parameter must not be altered by the EXIT routine.

TSCLASS=
The SYSOUT class for the SYSTSPRT DD statement for submitted jobs. It should be different from the MSGCLASS parameter and the class specified for other SYSOUT data sets in the specified PROC.

MAXCONN=
The maximum number of open sockets at any one time. Usually, each client requires 2 sockets while the command is being processed and the output is being returned. The default is 512.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4404I **Parameters: MSGCLASS=*class*,TSCLASS=*class*, TSOPROC=*value*,MAXCONN=*value*, NOEXIT**

Explanation

Indicates the values that have been specified for the parameters of the RXSERVE cataloged procedure. The following list provides a description of these parameters:

EXIT=
Name of user exit routine to inspect and alter JOB and EXEC parameters prior to submission of TSO batch jobs initiated by remote commands. In this case, there is no user exit routine.

TSOPROC=
The name of the TSO batch procedure. The default is IKJACCNT. The name IKJACCNT can be modified in the exit routine specified with the EXIT parameter.

MSGCLASS=
The MSGCLASS parameter for TSO batch jobs submitted to execute remote commands. The parameter must not be altered by the EXIT routine.

TSCLASS=
The SYSOUT class for the SYSTSPRT DD statement for submitted jobs. It should be different from the MSGCLASS parameter and the class specified for other SYSOUT data sets in the specified PROC.

MAXCONN=
The maximum number of open sockets at any one time. Usually, each client requires 2 sockets while the command is being processed and the output is being returned. The default is 512.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4405E **rexecd: command line exceeds 246 character limit; request denied.**

Explanation

The remote execution server has submitted a job to the MVS system. However, because the command line is longer than 246 characters, the job will fail with a JCL error. The server will not process the requested command.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Decrease the size of the command to 246 characters or less.

Module

MVRSHD

Procedure name

main

EZA4406I **rexecd: Too many matching jobnames exist for output to be returned.**

Explanation

An REXEC command was issued to an MVS REXEC server requesting remote execution. Jobnames are formed by concatenating 4 fixed character bytes with the socket number the request was presented on via the REXEC command. When an attempt was made to return the output, a request for a list of matching jobnames was specified. The data area for this request was too small to contain the list of all jobs matching the specification. Output from the job remains on the output spool.

System action

Processing continues.

Operator response

Determine which jobnames are duplicated and purge the duplicate job output.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main.

EZA4407I **Too many matching jobnames exist for output to be purged.**

Explanation

An REXEC command was issued to an MVS REXEC server requesting remote execution. Jobnames are formed by concatenating 4 fixed character bytes with the socket number the request was presented on via the REXEC command. When an attempt was made to purge the output, a request for a list of matching jobnames was specified. The data area for this request was too small to contain the list of all jobs matching the specification. Output from the job remains on the output spool.

System action

Processing continues.

Operator response

Determine which jobnames are duplicated and purge the duplicate job output.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main.

EZA4413E **rexecd: non-valid parameter JES=*x* specified JES=2 or JES=3 must be specified.**

Explanation

JES= was specified in the parameter list to the RECEC server. The JES= parameter only permits JES=2 or JES=3 to be specified.

System action

The REXEC server ends execution.

Operator response

Restart the REXEC server specifying JES=2 if your facility uses JES2, or JES=3 if your facility uses JES3.

System programmer response

Verify the JES= parameter is correct.

Module

RSHD

EZA4414I **rexecd: Initialization using JES=X selected by Y**

Explanation

This message indicates how the JES= parameter was determined. It could have been specified in the parameter or it could have been automatically detected.

X will be 2 on JES2 or 3 on JES3 systems.

Y will be:

- *Automatic Specification* if JES=x was not specified in the parameters.
- *Parameter Specification* if JES=x was specified in the parameters.

System action

The REXEC server continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSHD

EZA4416I **rexecd: Required resources are unavailable to process your request. Please try again later.**

Explanation

There are currently 9999 rexec jobs running. REXECD cannot submit any more jobs.

System action

Processing continues.

Operator response

Wait and then reissue the REXEC request.

System programmer response

No action is needed.

Module

MVUSHD

Procedure name

main

EZA4418I **rexecd: Required resources are unavailable to process request for *userid*.**

Explanation

There are currently 9999 rexec jobs running. REXECD cannot submit any more jobs.

userid is the user ID of the submitter of the request.

System action

Processing continues.

Operator response

Wait and then reissue the REXEC request.

System programmer response

No action is needed.

Module

MVUSHD

Procedure name

main

EZA4419I **rexecd: *function* rhosts = *fd***

Explanation

This is issued when trace is turned on.

function is the function being performed on the rhost file.

fd is the file descriptor of the rhost file.

System action

Processing continues.

Operator response

Wait and then reissue the REXEC request.

System programmer response

No action is needed.

Module

MVUSHD

Procedure name

main

EZA4420E **rexecd: Jobname was Not Found.**

Explanation

The server has detected the situation where a job that it was waiting for cannot be found.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main.

EZA4421E **chkpass failed: safrc=*rcode1*, racfrc=*rcode2*, racfre=*reason***

Explanation

The call to the chkpass function failed. Chkpass is called to validate a user ID and password with the system authorization facility (SAF). *rcode1* is the system authorization facility return code, *rcode2* is the RACF return code, and *reason* is the RACF reason code. This message is only written to the trace log.

System action

Message EZA4386E will be sent to the RSH client or message EZA4422E will be sent to the REXEC client. The connection to the client will be closed.

Operator response

See response for message EZA4386E if using RSH or EZA4422E if using REXEC.

System programmer response

Check the system authorization facility documentation to determine the cause of the failure. If RACF is installed, check the return codes for RACROUTE REQUEST=VERIFY in the z/OS Security Server RACROUTE Macro Reference.

Module

MVRSHD

Procedure name

main

EZA4422E **rexecd: Permission Denied.**

Explanation

Either the user ID or the password specified for the user ID is not valid on the remote system.

System action

Processing continues.

Operator response

Check the -l and -p parameters on REXEC and make sure that they are correct. If the user ID or the password, or both are being retrieved from a NETRC file, check the user ID and password in the NETRC file.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

main

EZA4423I **Parameters: PURGE = *purge*,PREFIX = *prefix*, IPV6=*ipv6*, SECLABEL = *seclabel*, PHRASE = *phrase***

Explanation

This message is a continuation of EZA4403I or EZA4404I. It indicates the values that have been specified for the parameters of the RXSERVE cataloged procedure.

purge
The values for purge are **Y** or **N**. **Y** indicates the job output from the jobs submitted by the server should be purged immediately after the job execution and **N** indicates that the job output will be held in the output queue.

prefix
A four-character value to be used as the first four characters in the jobname of the jobs that are submitted by the server. The remaining characters of the jobname are a sequential number between 1 and 9999.

ipv6
The values for *ipv6* are Y or N. If N is specified, RXSERVE will not attempt to communicate over an IPv6 network. Only connections from IPv4 clients will be accepted. If this parameter is not specified or Y is specified, RXSERVE will attempt to communicate over an IPv6 network.

seclabel
The values for seclabel are Y or N. If this parameter is not specified or N is specified, RXSERVE does not attempt to add a security label to the job card. If Y is specified, RXSERVE adds a security label (if one exists) to the job card following the message class parameter. For information about using the remote execution server in an MLS environment, see the section on preparing for TCP/IP networking in a multilevel security environment in z/OS Communications Server: IP Configuration Guide.

phrase
The values for phrase are Y or N. If this parameter is not specified or N is specified, RXSERVE supports a password size of 16 characters. If Y is specified, RXSERVE supports a password phrase size of 102 characters when the phrase is enclosed in single quotation marks or 100 characters without single quotation marks.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4424E **PURGE = *value* is not valid**

Explanation

This message indicates that an invalid value was specified for the PURGE option.

value is the value that was incorrectly specified.

System action

Processing continues. The option is ignored.

Operator response

Reissue the option with a correct value.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4425E **PREFIX parameter may not be set after START.**

Explanation

A PREFIX parameter was specified with a MODIFY command after the server had been started. PREFIX specifies the first four characters of a job submitted by the server. PREFIX must be specified when the server is started.

System action

Processing continues with the value declared for the PREFIX parameter at the time of initialization.

Operator response

If new PREFIX value is required then stop the server and start it again with the new PREFIX value.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4426E **Load of user exit *user_exit* failed**

Explanation

The remote execution server was unable to load the user exit specified on the EXIT= start parameter.

user_exit is the start parameter specified.

System action

Processing continues with no user exit in effect.

Operator response

Contact the Systems Programmer.

System programmer response

The remote execution server will issue a fetch() command to load the user exit. This message indicates that the fetch() command failed. See the z/OS XL C/C++ Runtime Library Reference for more information about the fetch() command.

Module

MVRSHD

Procedure name

process_options

EZA4427I **User exit *user_exit* loaded at *address***

Explanation

This message indicates that *user_exit* was successfully loaded.

user_exit is the specified start parameter.

address is the hexadecimal address where *user_exit* is located.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4428I **rhosts input: local host *host* local user *user***

Explanation

This message is issued by the MVS remote execution server when attempting to determine if the user is permitted to execute commands on the remote host. Local host and local user indicate what is being searched for in the *userid*.rhosts.data file.

host is the name of the local host.

user is the local user ID.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

doit

EZA4430E **No matching entry in *userid*.rhosts.data - user permission is denied**

Explanation

This message is issued when no matching entry is found in the *userid*.rhosts.data file. See message EZA4428I for an indication of the search values.

userid is the local user ID.

System action

Processing continues but permission for command execution is denied to the user.

Operator response

No action is needed.

System programmer response

Ensure that the rhosts.data file contains correct entries. For a description of the rhosts.data file format, see the information about remote execution in z/OS Communications Server: IP Configuration Guide.

Module

MVRSHD

Procedure name

doit

EZA4432E **IPV6 = value is not valid.**

Explanation

This message indicates that an invalid value was specified for the IPv6 option.

value is the invalid value specified.

System action

Processing continues. The option is ignored.

Operator response

Reissue the option with a valid value.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4433I **SECLABEL = *value* is not valid**

Explanation

This message indicates that an invalid value was specified for the SECLABEL option.

value is the invalid value specified.

System action

Processing continues. The option is ignored.

Operator response

Reissue the option with a valid value. Restart RXSERVE.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4434I **rexecd: Number of available job numbers is being depleted**

Explanation

REXECD is capable of tracking 9999 jobs. This message is issued when the number of jobs being tracked is 85% of this limit.

System action

A job number is assigned to the request and processing continues. This message is issued only once.

Operator response

Issue a modify command to change the PURGE parameter value to Y.

System programmer response

If the server is running with the PURGE parameter value N, remove jobs that have already been processed from the JES spool. Change the PURGE parameter value to Y in the remote execution cataloged procedure and recycle REXECD. Or use a modify command to change the PURGE parameter value to Y and add PURGE=Y to the remote execution cataloged procedure.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

This message is output to the console.

Example

Not applicable.

EZA4435E **rexecd: Number of available jobs is depleted**

Explanation

REXECD is capable of tracking 9999 jobs. This message is issued when the number of jobs that are being tracked has reached this limit.

System action

This message is issued once when REXECD detects that all available job numbers have been used. REXECD attempts to identify a job number that has been assigned but has no output on the JES spool.

- If a job number with no output on the JES spool is found, that job number is re-assigned and processing continues.
- If no job number can be assigned, the request is not processed.

Operator response

Issue a modify command to change the PURGE parameter value to Y.

System programmer response

If the server is running with the PURGE parameter value N, remove jobs that have already been processed from the JES spool. Change the PURGE parameter value to Y in the remote execution cataloged procedure and recycle REXECD. Or use a modify command to change the PURGE parameter value to Y and add PURGE=Y to the remote execution cataloged procedure.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

1

Descriptor code

2

Automation

This message is output to the console.

Example

Not applicable.

EZA4436I **SSCSARAY[*n*] *jobnumber jobstatus* ACTIVE**

Explanation

This is a trace message that is issued to indicate that the job is active.

In the message text:

n
The internal array index for the job number.

jobnumber
The job number of the job that is being tracked.

jobstatus
The status returned by JES, which indicates that the job is currently active.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA4436I SSCSARAY[0] JOB00131 80 ACTIVE
```

EZA4437I **SSCSARAY[*n*] *jobnumber jobstatus* WAITING**

Explanation

This is a trace message that is issued to indicate that the job is on the JES input queue and is waiting to be started.

A job might be in this state for the following reasons:

- A JES initiator was not started to process the submitted job class.
- Other jobs might be running in this class that are preventing this job from starting.

In the message text:

n
The internal array index for the job.

jobnumber
The job number of the job that is being tracked.

jobstatus
The status returned by JES, which indicates the job is waiting.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA4437I SSCSARAY[0] JOB00131 40 WAITING
```

EZA4438I **SSCSARAY[*n*] *jobnumber jobstatus* COMPLETED**

Explanation

This is a trace message that is issued to indicate that the job has completed processing.

In the message text:

n
The internal array index for the job.

jobnumber

The job number of the job that is being tracked.

jobstatus

The status returned by JES, which indicates the job has completed processing.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA4439I SSCSARAY[0] JOB00131 20 COMPLETED
```

EZA4439I **SSCSARAY[*n*] *jobnumber jobstatus* HELD**

Explanation

This is a trace message that is issued to indicate that the job is on the JES spool and is being held and is not available for processing.

In the message text:

n
The internal array index for the job.

jobnumber
The job number of the job that is being tracked.

jobstatus
The status returned by JES, which is indicates that the job is being held.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA4439I SSCSARAY[0] JOB00131 10 HELD
```

EZA4440I **Closing connection with *remoteaddress* from *localaddress***

Explanation

The connection to the remote IP address from the local IP address is being closed.

In the message text:

remoteaddress
The remote IP address and port number.

localaddress
The local IP address and port number.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RSHD

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
EZA4440I Closing connection with  9.44.36.163:1023 from 9.42.105.158:514
```

EZA4442I SSORT(next): 00000004 NO MORE OUTPUT ON JES SPOOL

Explanation

While processing the output for a job, JES returned a 00000004 return code. This trace message is issued to indicate that there is no more output on the JES spool for the specified job. Message EZA4392I that was issued earlier for the same socket identified the job that was being processed.

System action

Processing continues.

Operator response

Not applicable.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

MVRSHD

Routing code

10

Descriptor code

12

Automation

Not applicable.

Example

```
socket 4: EZA4392I S99ret: 00000000, T RSHD USER1.RSHD3.JOB00049.D0000102.?

socket 4: EZA4393I S99ret: 00000000

socket 4: EZA4442I SSSORT(next): 00000004 NO MORE OUTPUT ON JES SPOOL
```

EZA4443I **Attempt to open *userid*.RHOSTS.DATA requested by *user* on *host***

Explanation

MVRSHD server detected an RSH client attempting to use *userid*.RHOSTS.DATA for authentication.

In the message text:

userid
The remote user ID specified by the RSH client to use for execution of the remote command

user
The local user ID of the RSH client

host
: Fully qualified local host name of the RSH client

System action

The system continues processing.

Operator response

Not applicable.

System programmer response

Not applicable.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS CS TCP/IP: REXEC

Module

MVRSHD

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
socket 2: EZA4443I Attempt to open USER1.RHOSTS.DATA requested by user2 on sys209.pok.im.com
```

EZA4444E **PHRASE = *value* is not valid.**

Explanation

This message indicates that an invalid value was specified for the PHRASE option.

value is the invalid value specified.

System action

Processing continues. The option is ignored.

Operator response

Reissue the option with a valid value.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4445E **PHRASE parameter may not be set after START.**

Explanation

The PHRASE parameter was specified with a MODIFY command after the server had already been started. PHRASE specifies the support for 100 byte password values. The default is N, limiting passwords to 16 characters. PHRASE must be specified when the server is started.

System action

Processing continues with the value declared for the PHRASE parameter at the time of initialization.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVRSHD

Procedure name

process_options

EZA4726I **Usage: rexec -? -d -b *tab* -l *usr* -p *pwd* -n -m -s *port* -t *fn* fhost cmd**

Explanation

This message is the first message of a group of messages that is posted when REXEC is entered without any parameters, with the -? parameter, or under certain conditions that are not valid.

System action

REXEC ends.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TSOREXEC

Example

Following is the output when executing rexec -?:

```
EZA4726I Usage: rexec  -? -d -b <tab> -l <usr> -p <pwd> -n -s <port> -t <fn> Fhost cmd
EZA4727I  options: -
EZA4728I           -?          display this message.
EZA4729I           -d          turn on debug tracing.
EZA4814I           -m          prefix '09'x to machine control output.
EZA4811I           -b <tab>    specifies tab value,
EZA4813I                       valid range 1 - 12 (default 1).
EZA4730I           -l <usr>    specifies remote userid.
EZA4731I           -p <pwd>    specifies remote password.
EZA4732I           -n          prevent automatic login.
EZA4733I           -s <port>   specifies server port (default 512).
EZA4802I           -t <fn>     specifies translation table name.

EZA4734I  Example: rexec  -d -l guest -p guest hostname ls
```

Procedure name

usage

EZA4727I **options: -**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4728I **-? display this message.**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4729I **-d turn on debug tracing.**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4730I **-l *usr* specifies remote userid.**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4731I **-p *pwd* specifies remote password.**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4732I **-n prevent automatic login.**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4733I **-s *port* specifies server port (default 512)**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4734I **Example: rexec -d -l guest -p guest hostname ls**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4736I **close error con # *errconn***

Explanation

An error has occurred on the port listed and the port is being closed.

System action

REXEC ends.

Operator response

Message EZA4779I precedes this message and explains what needs to be corrected prior to rerunning the job.

Note: This is a normal message if the foreign host closed the connection, as shown by CONNECTIONstateCHANGED to sending prior to receiving this message.

System programmer response

No action is needed.

Module

REXEC

Procedure name

close

EZA4737I **close active con # *actconn***

Explanation

REXEC is in the process of ending and the port is being closed.

System action

REXEC ends.

Operator response

Message EZA4779I precedes this message and explains what needs to be corrected prior to rerunning the job.

System programmer response

No action is needed.

Module

REXEC

Procedure name

close

EZA4738I **getnextnote until DD**

Explanation

The program has entered the procedure recv_notices.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices

EZA4739I ***SayNotEn(string)***

Explanation

This message ID can have any one of several messages. A listing of the messages and their descriptions follows. The operator and system response will vary depending on what the message is. Look at the message and the description to determine the action to take. If you still do not understand the problem contact the IBM Support Center for assistance.

Buffer space available (8680)

This notification is given when buffer space becomes available for a connection on which a TCP send request was previously rejected due to insufficient resources.

Connection state changed (8681)

This notification is given when a TCP connection receives information from the network that causes the state of the connection to change. Such changes arise when the remote client opens, closes, or ends the connection.

Data delivered (8682)

This notification is given when the client buffer indicated in an earlier TcpReceive request now contains data.

User-defined notification (8683)

For use by user-written programs.

Datagram space available (8684)

This notification is given when buffer space becomes available for a datagram on which an IP send request was previously rejected due to insufficient resources.

Urgent pending (8685)

This notification is given when the TCP/IP service is informed by the foreign site that there is urgent data not yet delivered to the client.

UDP datagram delivered (8686)

This notification is given when the client buffer indicated in an earlier UdpReceive request now contains a datagram.

UDP datagram space available (8687)

This notification is given when buffer space becomes available for a datagram on which an UDP send request was previously rejected due to insufficient resources.

Other external interrupt received (8688)

This special notification is returned when a non-VMCF external interrupt arrives for the client. Typically, this external interrupt will be CPexternalCOMMAND and will be interpreted by the client as a directive to abend.

User delivers line (8689)

The user has entered a line of data at the terminal.

User wants attention (8690)

The user has pressed an attention key, such as PF key.

Timer expired (8691)

A timer has expired.

Fsend response (8692)

Response has been received for TcpFSend call.

FReceive error (8693)

This notification is given when a TcpFReceive call resulted in an error, instead of a DATAdelivered notification. Every TcpFReceive call will eventually generate either DATAdelivered or RECEIVEerror.

RawIP packets delivered (8694)

This notification is given when the client buffer indicated in an earlier RawIpReceive request now contains data.

RawIP space available (8695)

This notification is given when buffer space becomes available for a datagram on which a RawIpSend request was previously rejected due to insufficient resources.

IUCV interrupt (8696)

This special notification is returned when an IUCV external interrupt arrives for the client.

I/O interrupt (8697)

This notification is given when an I/O interrupt occurs on a device that the client program has specified in a NotifyIo() call.

Resources available for TcpOpen (8698)

This notification is given to a client who previously tried to do TcpOpen but got ZEROresources. When all resources necessary for open are available (TCB and SCB plus incoming and outgoing data buffer), this notice is given.

Resources available for UdpOpen (8699)

Similar to RESOURCESavailable, for TcpOpen.

Ping response or timeout (8700)
Sent when a PING response is received, or when the request times out.

SMSG receive (8701)
Sent when a CP special message (SMSG) is received.

System action

Varies according to the message displayed.

Operator response

Varies according to the message displayed.

System programmer response

Varies according to the message displayed.

Module

REXEC

Procedure name

recv_notices

EZA4740I ***SayConSt(message_string)***

Explanation

This message ID can have any one of several *message_strings*. A list of *message_strings* and their descriptions follows. The operator and system programmer response will vary depending on what the *message_string* is. Look at the *message_string* and the description to determine the action to take. If you still do not understand the problem contact the IBM Support Center for assistance.

Connection closing (8670)
Data can no longer be transmitted on this connection since the TCP/IP service is in the process of closing down the connection.

Listening (8671)
Waiting for a foreign site to open a connection.

Nonexistent (8672)
The connection no longer exists.

Open (8673)
Data can go either way on the connection.

Receiving only (8674)
Data can be received but not sent on this connection because the client has done a one-way close.

Sending only (8675)
Data can be sent out but not received on this connection. This means that the foreign site has done a one-way close.

Trying to open (8676)
Trying to contact a foreign site to establish a connection.

**** Invalid state: ord=*ord(STATE)* (8679)**
The state was not one of the above.

System action

Varies according to message displayed.

Operator response

Varies according to the message displayed.

System programmer response

Varies according to the message displayed.

Module

REXEC

Procedure name

recv_notices.

EZA4742I **Unknown connection state change**

Explanation

The CONNECTIONstateCHANGED was not one of the following:

OPEN
RECEIVINGonly
LISTENING
TRYINGtoOPEN
SENDINGonly
CONNECTIONclosing
NONEXISTENT

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices

EZA4743I **Bytes in *bytes delivered***

Explanation

This message indicates the number of bytes that have been received from a TcpReceive. There will be 1 extra byte per line to account for the carriage return.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices

EZA4744I ***errmsg (msgnum)***

Explanation

REXEC encountered an error.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

REXEC

Procedure name

recv_notices

EZA4745I **tcpreceive error from errcon**

Explanation

This message will be preceded by message EZA4744I, which should be consulted to determine the error. This error message is printed when a TcpReceive has been issued and the return code was something other than OK, NOsuchCONNECTION, or REMOTEclose.

System action

REXEC continues.

Operator response

Varies depending on the message EZA4744I.

System programmer response

Varies depending on the message EZA4744I.

Module

REXEC

Procedure name

recv_notices.

EZA4746I **tcpreceive error from actconn**

Explanation

This message will be preceded by message EZA4744, which should be consulted to determine the error. This error message is printed when a TcpReceive has been issued and the return code was something other than OK, NOsuchCONNECTION, or REMOTEclose.

System action

REXEC continues.

Operator response

Varies depending on the message EZA4744I.

System programmer response

Varies depending on the message EZA4744I.

Module

REXEC

Procedure name

recv_notices.

EZA4748I **external interrupt *code* encountered**

Explanation

An external interrupt has been received. If an external interrupt is received, the port is closed in preparation for a shutdown.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices

EZA4749I **ignoring.**

Explanation

If the connection state is not one of the following: CONNECTIONstateCHANGED, DATAdelivered, or EXTERNALinterrupt, then it is ignored.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices

EZA4750I **unknown notification**

Explanation

The connection state message is not one that has been previously defined.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices.

EZA4751I **Returning from recv_notices**

Explanation

The DEBUG option has been specified and processing is complete for the procedure recv_notices.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

recv_notices

EZA4752I **Please enter command (or “quit” to exit)**

Explanation

A command needs to be entered for the remote host. If “quit” is entered, then REXEC will end.

System action

REXEC awaits input.

Operator response

Enter a command.

System programmer response

No action is needed.

Module

REXEC

Procedure name

prompt_for_command

EZA4753I **PASSWD (*fhost: userid*)**

Explanation

A password needs to be entered for the user ID on the remote host. If quit is entered, REXEC will end.

System action

REXEC awaits input.

Operator response

Enter the password.

System programmer response

No action is needed.

Module

REXEC

Procedure name

prompt_for_passwd

EZA4754I **USERID (*fhost*: userid)**

Explanation

A user ID needs to be entered for the remote host. If quit is entered REXEC will end.

System action

REXEC awaits input.

Operator response

Enter the user ID.

System programmer response

No action is needed.

Module

REXEC

Procedure name

prompt_for_userid

EZA4755I **Please enter host name:**

Explanation

A remote host name needs to be entered. If quit is entered REXEC will end.

System action

REXEC awaits input.

Operator response

Enter the remote host name.

System programmer response

No action is needed.

Module

REXEC

Procedure name

prompt_for_fhost

EZA4756I **Variables have the following assignments:**

Explanation

The DEBUG option has been specified and the variables that have been entered will be displayed in messages EZA4757I - EZA4760I.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

show_variables

EZA4757I **fhost: *fhost***

Explanation

The DEBUG option has been specified and the remote host is displayed. See message EZA4756I.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

show_variables

EZA4758I **userid: *userid***

Explanation

The DEBUG option has been specified and the logon ID is displayed. See message EZA4756I.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

show_variables

EZA4759I **passwd: *passwd***

Explanation

The DEBUG option has been specified and the password entered is displayed. See message EZA4756I.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

show_variables

EZA4760I **command: *cmd***

Explanation

The DEBUG option has been specified and the command entered is displayed. See message EZA4756I.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

show_variables

EZA4761I **ALL: *theline***

Explanation

This is an informational message that displays the line of the NETRC data set that is being read, if DEBUG has been specified.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4762I **MACHINE: *word***

Explanation

This message displays the word following MACHINE in the NETRC data set, if DEBUG has been specified.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4763I **LOGIN: *word***

Explanation

The word following LOGIN in the NETRC data set is displayed if DEBUG has been specified.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4764I **PASSWORD: *word***

Explanation

This is an informational message that displays the word following PASSWORD in the NETRC data set, if debug is set.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4765I **REJECTED: *word***

Explanation

This message will be printed if a word is found other than MACHINE and a machine name, LOGIN and login name, and PASSWORD and password. This message will be followed by the following:

EZA4766E
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

System action

REXEC ends.

Operator response

Correct the user_id.NETREC.DATA data set.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4766I **Invalid data set format in *netrc_data_set_name***

Explanation

During the parameter input data set read procedure, a format error was found in the *NETRC_data_set*. REXEC does not prompt you for the correct parameter input.

System action

REXEC displays the following group of messages and ends:

EZA4766I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Correct the *NETRC_data_set*, and restart the program. Otherwise set the -n option for prompted input, or use the execution invocation argument inline parameter input when you rerun the program.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4767I **-l option required one parameter&colon *alternate login id***

Explanation

A parameter for the -l option was missing or not correct.

System action

REXEC prints the following group of messages and ends.

EZA4767I

EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Reenter the command with a correct logon ID.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4768I **-p option requires one parameter&colon *passwd***

Explanation

A parameter for the -p option was missing or not correct.

System action

REXEC prints the following group of messages and ends.

EZA4768I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Reenter the command with a valid password.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4769I **-s option requires one parameter: *port***

Explanation

A parameter for the -s option was missing or not correct.

System action

REXEC prints the following group of messages and ends.

EZA4769I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Reenter the command with a valid port number.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4770I **Unknown option *word***

Explanation

A parameter was entered that could not be parsed.

System action

REXEC prints the following group of messages and ends:

EZA4770I
EZA4726I
EZA4727I

EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Reenter the command with valid parameters.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4771I **REXEC: No remote host name given.**

Explanation

The host name was not entered.

System action

REXEC prints the following group of messages and ends.

EZA4771I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Reenter the command with a valid host name.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4772I **parms are *parameters***

Explanation

When DEBUG is set on it will list the parameters that were passed to REXEC.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4773I **REXEC: no command given.**

Explanation

This message is displayed if no command was parsed.

System action

REXEC displays the following group of messages and ends.

```
EZA4773I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I
```

Operator response

Reenter the complete REXEC command.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4774I **REXEC invoked;**

Explanation

The DEBUG option has been entered and the first statement of procedure REXEC_util has been processed.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_util

EZA4775I **Calling function rexec_af with: Host: *aaa* user *bbb*, cmd *ccc*, port *ddd***

Explanation

This message lists the host, user, cmd, and port that are being passed to the rexec_af function that is provided by z/OS UNIX System Services and that does the work of passing the information to the remote server.

aaa is the host that the command 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.

ddd is the port number.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_util

EZA4776I **D2 len *length***

Explanation

This is the length of the string that is being sent to the remote host.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_util

EZA4777I **tcpwaitsend *errmsg (msgnum)***

Explanation

The function tcpwaitsend encountered an error.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

REXEC

Procedure name

REXEC_util

EZA4778I **returning from REXEC_UTIL**

Explanation

The DEBUG option has been entered and the last statement of procedure REXEC_util has been processed.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_util

EZA4779I ***errmsg (msgnum)***

Explanation

REXEC encountered an error.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

REXEC

Procedure name

REXEC_Client

EZA4780I **calling GetHostResol with *fhost***

Explanation

REXEC is calling GetHostResol to resolve the remote host name to a dotted decimal address.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_Client

EZA4781I **Host *fhost* Unknown**

Explanation

The call to GetHostResol has not resolved the name to a dotted decimal address.

System action

REXEC ends.

Operator response

Reenter the command using a valid remote host name.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_Client

EZA4783I **Passive Conn – *errmsg (msgnum)* on local port *localsocket.port***

Explanation

If everything is working correctly, Passive Conn – OK on local port *port number* will be displayed.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

REXEC

Procedure name

REXEC_Client

EZA4784I **Unable to open passive port for host *host***

Explanation

This message precedes message EZA4779I. Consult that message for an explanation of why the port was unable to be opened.

System action

REXEC ends.

Operator response

Varies depending on the message displayed.

System programmer response

Varies depending on the message displayed.

Module

REXEC

Procedure name

REXEC_Client

EZA4785I **passive open complete on port *connection***

Explanation

A passive open has been completed and the connection state has changed to LISTENING.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_Client

EZA4786I **Active Conn – *errmsg (msgnum)* on local port *localsocket.port***

Explanation

If everything is working correctly, Active Conn – OK on local port *port number* will be displayed.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

REXEC

Procedure name

REXEC_Client

EZA4787I **Unable to open port *TCP_REXEC* to host *fhost***

Explanation

This message is preceded by message EZA4779I. Please consult that message for an explanation of why the port was unable to be opened.

System action

REXEC ends.

Operator response

Varies depending on the message displayed.

System programmer response

Varies depending on the message displayed.

Module

REXEC

Procedure name

REXEC_Client

EZA4788I **active open complete on port *connection***

Explanation

An active open is complete.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_Client

EZA4789I **REXEC complete**

Explanation

The REXEC program has ended.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_Client

EZA4790I **syntax error: unbalanced quotation marks *substr(line,position)***

Explanation

This message is displayed if a quoted string has been entered and there are unbalanced quotation marks. This message will be followed by the following group of messages:

EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

System action

REXEC ends.

Operator response

Correct the quoted string and reenter the command.

System programmer response

No action is needed.

Module

REXEC

Procedure name

TcpQuote

EZA4791I **Connecting to *fhost***

Explanation

The REXEC is attempting to connect to the foreign host name, the foreign host address, and the remote host port given in the message.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

REXEC_Client

EZA4801I **MVS TCP/IP REXEC *version***

Explanation

The version of REXEC is displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4802I **-t *fn* specifies translation table name.**

Explanation

This is one of a group of messages that is posted when REXEC is entered without any parameters, or with the -? parameter, or under certain conditions that are not valid. The first message is EZA4726I. See that message for a complete description of the group.

System action

REXEC ends.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4803I **-t option requires one parameter**

Explanation

A parameter for the -t option was missing or not valid.

System action

REXEC prints the following group of messages and ends:

EZA4803I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Correct or enter the required parameter.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4804I **User specified translation table: *table***

Explanation

The user-specified translation table has been read in and will be used.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4805I **Fails to read User specified translation table: *table***

Explanation

The translate table that was supplied by the user could not be initialized.

System action

REXEC ends.

Operator response

Verify that the data set that has been specified with the -t option is correctly named and available to the system.

See the information about understanding configuration search orders in z/OS Communications Server: IP Configuration Guide for information about REXEC's data set search order.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4806E **Invalid integer parameter: *word***

Explanation

The port being read in with the -s option is non-numeric.

System action

REXEC prints the following group of messages and ends.

EZA4806I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Correct the port parameter.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4807E **Invalid port number *number***

Explanation

The port number read in was not between the range of 1 and 65535.

System action

REXEC prints the following group of messages and ends:

EZA4807I
EZA4726I
EZA4727I
EZA4728I
EZA4729I
EZA4730I
EZA4731I
EZA4732I
EZA4733I
EZA4802I
EZA4734I

Operator response

Correct the port number.

System programmer response

No action is needed.

Module

REXEC

Procedure name

GETPARM

EZA4808E **Fails to read standard table**

Explanation

An REXEC command was issued from an MVS REXEC client requesting remote execution. The standard translation table used by the client was found; however, it could not be successfully used.

System action

TCPIP continues. The REXEC client command terminates.

Operator response

Reissue the REXEC command. If the problem persists, seek assistance from appropriate system support personnel.

System programmer response

No action is needed.

Module

EZABBO1T

Procedure name

Main.

EZA4809I **Using NETRC file *NETRC_data_set***

Explanation

The debug option has been specified and the NETRC data set being used is displayed.

System action

REXEC continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

REXEC

Procedure name

from_netrc

EZA4810E **The call to rexec_af() function failed: *description* rsn = *errnojr***

Explanation

The call to the rexec_af() function that will pass the information to the remote host failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System action

REXEC ends.

Operator response

Examine the command typed and reissue the REXEC command. If the problem persists, contact the system programmer. The most common description is ECONNREFUSED. If this occurs, start the remote execution server on the remote host.

System programmer response

Correct the error indicated by *description* and *errnojr*.

Module

REXEC

Procedure name

main()

EZA4811I **-b *tab* specify tab value**

Explanation

This message is part of message group for REXEC.

In the message text:

tab
The fixed tab spacing used to align data when the command response contains a tab character. The *tab* value is in the range of 1-12.

System action

The default value of *tab* is 1.

Operator response

Enter a valid value.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

EZAREXC

Routing code

10

Descriptor code

12

Example

```
EZA4726I Usage: rexec  -? -d -b tab -l usr -p pwd -n -s port -t fn Fhost cmd
EZA4727I  options: -
EZA4728I           -?          display this message.
EZA4729I           -d          turn on debug tracing.
EZA4811I           -b tab    specifies tab value
EZA4813I                       valid range 1 - 12, (default 1).
EZA4730I           -l usr    specifies remote userid.
EZA4731I           -p pwd    specifies remote password.
EZA4732I           -n          prevent automatic login.
EZA4733I           -s port   specifies server port (default 512.)
EZA4802I           -t fn     specifies translation table name.
EZA4734I  Example: rexec  -d -l guest -p guest hostname ls
```

EZA4812I **A tab value of *tabcol* is not in the range of 1 - 12**

Explanation

The value entered for the tab parameter is outside the valid range.

In the message text:

tabcol
The column width specified for the tab.

System action

The REXEC client exits.

Operator response

Enter a valid value and restart the client.

System programmer response

Not applicable.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

EZAREXC

Routing code

10

Descriptor code

12

Example

```
EZA4812I A tab value of 13 is not in the range of 1 - 12
```

EZA4813I **valid range 1 thru 12 (default 1)**

Explanation

This message is a continuation of the EZA4811I message

System action

See message EZA4811I.

Operator response

See message EZA4811I.

System programmer response

See message EZA4811I.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

EZAREXC

Routing code

10

Descriptor code

12

Example

See message EZA4811I.

EZA4814I **-m prefix '09'x to machine control output.**

Explanation

This message is part of a group of messages. The first message is EZA4726I. See that message for a complete description of the group.

EZA4986I **close error con # *connection***

Explanation

An error has occurred on the port listed and the port is being closed.

System action

RSH ends.

Operator response

Message EZA5029I precedes this message and explains the correction that needs to be made prior to rerunning the job.

System programmer response

No action is needed.

Module

RSH

Procedure name

close, recv_notices

EZA4987I **close active con # *connection***

Explanation

RSH is in the process of ending and the port is being closed.

System action

RSH ends.

Operator response

Message EZA5029I precedes this message and explains the correction that needs to be made prior to rerunning the job.

Note: This is a normal message if the foreign host closed the connection, as shown by CONNECTIONstateCHANGED to sending prior to receiving this message.

System programmer response

No action is needed.

Module

RSH

Procedure name

close

EZA4988I **getnextnote until DD**

Explanation

The program has entered the procedure recv_notices.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

EZA4989I ***SayNotEn(string)***

Explanation

This message ID can have any one of several messages. A listing of the messages and their meanings follows. The operator and system programmer response will vary depending on the message displayed. If you still do not understand the problem, contact the IBM Software Support Center for assistance.

Message
Explanation

Buffer space available (8680)
This notification is given when buffer space becomes available for a connection on which a TCP send request was previously rejected due to insufficient resources.

Connection state changes (8681)
This notification is given when a TCP connection receives information from the network that causes the state of the connection to change. Such changes arise when the remote client opens, closes, or ends the connection.

Data delivered (8682)
This notification is given when the client buffer indicated in an earlier TcpReceive request now contains data.

User-defined notification (8683)
For use by user-written programs.

Datagram space available (8684)
This notification is given when buffer space becomes available for a datagram on which an IP send request was previously rejected due to insufficient resources.

Urgent pending (8685)
This notification is given when the TCPIP service is informed by the foreign site that there is urgent data not yet delivered to the client.

UDP datagram delivered (8686)
This notification is given when the client buffer indicated in an earlier UdpReceive request now contains a datagram.

UDP datagram space available (8687)
This notification is given when buffer space becomes available for a datagram on which an UDP send request was previously rejected due to insufficient resources.

Other external interrupt receive (8688)
This special notification is returned when a non-VMCF external interrupt arrives for the client. Typically, this external interrupt will be CPexternalCOMMAND, which will be interpreted by the client as a directive to abend.

User delivers line (8689)
The user has entered a line of data at the terminal.

User wants attention (8690)
The user has pressed an attention key, such as a PF key.

Timer expired (8691)
A timer has expired.

Fsend response (8692)
Response has been received for TcpFSend call.

FReceive error (8693)
This notification is given when a TcpFReceive call resulted in an error, instead of a DATAdelivered notification. Every TcpFReceive call will eventually generate either DATAdelivered or RECEIVEerror.

RawIP packets delivered (8694)
This notification is given when the client buffer indicated in an earlier RawIpReceive request now contains data.

RawIP space available (8695)
This notification is given when buffer space becomes available for a datagram on which a RawIpSend request was previously rejected due to insufficient resources.

IUCV interrupt (8696)
This special notification is returned when an IUCV external interrupt arrives for the client.

I/O interrupt (8697)
This notification is given when an I/O interrupt occurs on a device that the client program has specified in a NotifyIo() call.

Resources available for TcpOpen (8698)
This notification is given to a client who previously tried to do TcpOpen but got ZEROresources. When all resources necessary for open are available (TCB and SCB plus incoming and outgoing data buffer), this notice is given.

Resources available for UdpOpen (8699)
Similar to RESOURCESavailable for TcpOpen.

Ping response or time out (8700)
Sent when a PING response is received, or when the request times out

SMSG received (8701)
Sent when a CP special message (SMSG) is received.

System action

Varies according to the message displayed.

Operator response

Varies according to the message displayed.

System programmer response

Varies according to the message displayed.

Module

RSH

Procedure name

recv_notices

EZA4990I ***SayConSt(string)***

Explanation

This message ID can have any one of several messages. A listing of the message and their meanings follows. The operator and system programmer response will vary depending on what the message is. Look at the message and the description to determine the action to take. If you still do not understand the problem contact the IBM Software Support Center for assistance.

Message
Explanation

Connection closing
Data can no longer be transmitted on this connection since the TCP/IP service is in the process of closing down the connection.

Listening (8671)
Waiting for a foreign site to open a connection.

Nonexistent (8672)
The connection no longer exists.

Open (8673)
Data can go either way on the connection.

Receiving only (8674)
Data can be received but not sent on this connection, because the client has done a one-way close.

Sending only (8675)
Data can be sent out but not received on this connection. This means that the foreign site has done a one-way close.

Trying to open (8676)
Trying to contact a foreign site to establish a connection.

**** Invalid state: ord = *ord(STATE)* *** (8679)**
The state was not one of the above.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Varies depending on the message displayed.

Module

RSH

Procedure name

recv_notices

EZA4992I **Unknown connection state change**

Explanation

The CONNECTIONstateCHANGED was not one of the following:

OPEN
RECEIVINGonly
LISTENING
TRYINGtoOPEN
SENDINGonly
CONNECTIONclosing
NONEXISTENT

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

EZA4993I **Bytes in *bytes***

Explanation

This message indicates the number of bytes that have been received from a TcpReceive. There will be 1 extra byte per line to account for the carriage return.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

EZA4994I ***errmsg (msgnum)***

Explanation

RSH encountered an error.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

RSH

Procedure name

recv_notices, rsh_util, GETPARM

EZA4995I **tcpreceive error from errcon**

Explanation

This message is preceded by message EZA4994I, which should be consulted to determine the error. This error message is displayed when a TcpReceive has been issued and the return code was something other than OK, NosuchCONNECTION, or REMOTEclose.

System action

RSH continues.

Operator response

Varies depending on message EZA4994I.

System programmer response

Varies depending on message EZA4994I.

Module

RSH

Procedure name

recv_notices

EZA4996I **tcpreceive error from actconn**

Explanation

This message will be preceded by message EZA4994I, which should be consulted to determine the error. This error message is printed when a TcpReceive has been issued and the return code was something other than OK, NOsuchCONNECTION, or REMOTEclose.

System action

RSH continues.

Operator response

Varies depending on information displayed in message EZA4994I.

System programmer response

Varies depending on information displayed in message EZA4994I.

Module

RSH

Procedure name

recv_notices, Rsh_Client

EZA4998I **external interrupt *code* encountered**

Explanation

An external interrupt has been received. If an external interrupt is received, the port is closed in preparation for a shutdown.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

EZA4999I **ignoring**

Explanation

If the connection state is not one of the following: CONNECTIONstateCHANGED, DATAdelivered, or EXTERNALinterrupt, then it is ignored.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

Chapter 7. EZA5*xxxx* messages

EZA5000I **unknown notification**

Explanation

The connection state message is not one that has been previously defined.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

EZA5001I **Returning from recv_notices**

Explanation

The DEBUG option has been specified, and processing is complete for the procedure recv_notices.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

recv_notices

EZA5006I **Variables have the following assignment:**

Explanation

The DEBUG option has been specified and the variables that have been entered will be displayed in the following messages.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

show_variables

EZA5007I **FHOST: *fhost***

Explanation

The DEBUG option has been specified and the foreign host is displayed.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

show_variables

EZA5008I **USERID: *userid***

Explanation

The DEBUG option has been specified and the logon ID is displayed.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

show_variables

EZA5010I **COMMAND: *cmd***

Explanation

The DEBUG option has been specified and the command entered is displayed.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

show_variables

EZA5017I **-l option requires one parameter: *alternate login id***

Explanation

A parameter for the -l option was missing or not correct.

System action

RSH displays the following group of messages and ends:

```
EZA5017I
EZA5042I
EZA4977I
```

EZA4978I
EZA4979I
EZA4980I
EZA5050I
EZA5043I

Operator response

Reenter the command with a valid logon ID.

System programmer response

No action is needed.

Module

RSH

Procedure name

GETPARM

EZA5019I **-s option requires one parameter: *port***

Explanation

A parameter for the -s option was missing or not valid.

System action

RSH displays the following group of messages and ends:

EZA5019I
EZA5042I
EZA4977I
EZA4978I
EZA4979I
EZA4980I
EZA5050I
EZA5043I

Operator response

Reenter the command with a valid port number.

System programmer response

No action is needed.

Module

RSH

Procedure name

GETPARM

EZA5020I **Unknown option *word***

Explanation

A parameter was entered that could not be parsed.

System action

RSH displays the following group of messages and ends:

EZA5020I
EZA5042I
EZA4977I
EZA4978I
EZA4979I
EZA4980I
EZA5050I
EZA5043I

Operator response

Reenter the command with valid parameters.

System programmer response

No action is needed.

Module

RSH

Procedure name

GETPARM

EZA5022I **parms are *parameters***

Explanation

When DEBUG is set it will list the parameters that were passed to RSH.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

GETPARM

EZA5025I **Calling function rcmd_af with: Host: AAA user BBB, cmd CCC, port DDD**

Explanation

This message lists the host, user, cmd, and port that are being passed to the local rcmd_af function which does the work of passing the information to the remote server. AAA is the host that the command 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

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

rsh_util

EZA5026I **D2 len *length***

Explanation

This is the length of the string that is being sent to the foreign host.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

rsh_util

EZA5027I **tcpwaitsent *errmsg (msgnum)***

Explanation

The function tcpwaitsend encountered an error.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

RSH

Procedure name

rsh_util

EZA5029I ***errmsg (msgnum)***

Explanation

RSH encountered an error.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

RSH

Procedure name

Rsh_client

EZA5030I **calling GetHostResol with *fhost***

Explanation

RSH is calling GetHostResol to resolve the foreign host name to a dotted decimal address.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

Rsh_client

EZA5031I **Host *fhost* Unknown**

Explanation

The call to GetHostResol has failed to resolve the name to a dotted decimal address.

System action

RSH ends.

Operator response

Reenter the command using a valid foreign host name.

System programmer response

No action is needed.

Module

RSH

Procedure name

Rsh_client

EZA5033I **Passive Conn - *errmsg (msgnum)* on local port *localsocket.port***

Explanation

RSH encountered an error on the indicated connection.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

RSH

Procedure name

Rsh_client

EZA5034I **Unable to open passive port for host *fhost***

Explanation

This message is preceded by message EZA5029I, which should be consulted for an explanation of why the port was unable to be opened.

System action

RSH ends.

Operator response

Varies depending on the message displayed.

System programmer response

Varies depending on the message displayed.

Module

RSH

Procedure name

Rsh_client

EZA5035I **passive open complete on port *connection***

Explanation

A passive open is complete and the connection state has been changed to LISTENING.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

Rsh_client

EZA5036I **Active Conn - *errmsg (msgnum) on local port***

Explanation

RSH encountered an error on the indicated connection.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

Varies depending on the message displayed.

Operator response

Varies depending on the message displayed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

RSH

Procedure name

Rsh_client

EZA5037I **Unable to open port *port number* to host *fhost***

Explanation

This message is preceded by message EZA5029I. Consult this message for an explanation of why the port was unable to open.

System action

RSH ends.

Operator response

Varies depending on the message displayed.

System programmer response

Varies depending on the message displayed.

Module

RSH

Procedure name

Rsh_client

EZA5038I **active open complete on port *connection***

Explanation

An active open is complete.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

Rsh_client

EZA5041I **Connecting to *fhost***

Explanation

RSH is attempting to connect to the foreign host name, the foreign host address, and the foreign host port given in the message.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

Rsh_client

EZA5042I	**Usage: rsh -? -d -b *tab* -l *usr* -s *port* fhost cmd**
EZA4977I	**options: -**
EZA4978I	**-? display this message.**
EZA4979I	**-d turn on DEBUG tracing.**
EZA4980I	**-l *usr* specifies remote userid.**
EZA5050I	**-s *port* specifies server port (default 514)**
EZA5043I	**Example: rsh -d -l guest hostname is**

Explanation

This group of messages is displayed when RSH is entered without any parameters, or, with the -? parameter, or under certain conditions that are not correct.

System action

RSH ends.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

usage, GETPARM, Rsh_Client

EZA5044I	**rsh: No foreign host name given.**

Explanation

No host name was given.

System action

RSH prints the following group of messages and ends.

EZA5044I
EZA5042I

EZA4977I
EZA4978I
EZA4979I
EZA4980I
EZA5050I
EZA5043I

Operator response

Reenter the command with a valid host name.

System programmer response

No action is needed.

Module

RSH

Procedure name

GETPARM

EZA5045I **rsh: no command given.**

Explanation

No command was parsed.

System action

RSH displays the following group of messages and ends.

EZA5045I
EZA5042I
EZA4977I
EZA4978I
EZA4979I
EZA4980I
EZA5050I
EZA5043I

Operator response

Reenter the command with a valid host name.

System programmer response

No action is needed.

Module

RSH

Procedure name

GETPARM x

EZA5046I **RSH invoked**

Explanation

The DEBUG option has been entered and the first statement of procedure Rsh_util has been processed.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

rsh_util

EZA5047I **returning from RSH_UTIL**

Explanation

The DEBUG option has been entered and the last statement of procedure rsh_util has been processed.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

rsh_util

EZA5048I **RSH complete**

Explanation

The program has ended.

System action

RSH continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

Rsh_Client

EZA5049I **locuser *locuser***

Explanation

This message displays the local user ID.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RSH

Procedure name

show_variables

EZA5051E **The call to rcmd_af procedure failed: *description* rsn = *errnojr***

Explanation

The call to the rcmd_af procedure that passes the information to the remote host failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System action

RSH ends.

Operator response

Examine the command typed and reissue the RSH command. If the problem persists, contact the system programmer. The most common description is ECONNREFUSED. If this occurs, start the remote shell server on the remote host.

System programmer response

Correct the error indicated by *description* and *errnojr*.

Module

RSH

Procedure name

main()

EZA5058I **-b *tab* specify tab value**

Explanation

This message is part of a message group for RSH.

System action

The client exits.

Operator response

Enter a valid value.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RSH

Module

EZATRSH

Routing code

10

Descriptor code

12

Example

```
EZA4977I  options: -
EZA4978I           -?          display this message.
EZA4979I           -d          turn on debug tracing.
EZA5058I           -b tab    specifies tab value,
EZA5060I                       valid range 1 - 12, (default 1).
EZA4980I           -l usr    specifies remote userid.
EZA5050I           -s port   specifies server port (default 514.)
EZA5043I  Example: rsh  -d -l guest hostname ls
```

EZA5059I **A tab value of *tabcol*, is not in the range of 1 - 12.**

Explanation

The value entered for the tab parameter is outside the valid range.

In the message text:

tabcol
The column width specified for the tab parameter.

System action

The RSH client exits.

Operator response

Enter a valid value and restart the client.

System programmer response

Not applicable.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RSH

Module

EZATRSH

Routing code

10

Descriptor code

12

Example

```
EZA5059I A tab value of 13 is not in the range of 1 - 12
```

EZA5060I **valid range 1 thru 12 (default 1)**

Explanation

This message is a continuation of message EZA5058I.

System action

The client exits.

Operator response

Enter a valid value.

System programmer response

Not applicable.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: RSH

Module

EZATRSH

Routing code

10

Descriptor code

12

Example

```
EZA5060I          valid range 1 thru 12 (default 1)
```

EZA5061I **Using user-specified translation table: *hlq.mlq*.TCPXLBIN**

Explanation

The Debug option was specified and the REXEC client is using the user-specified translation table.

In the message text:

hlq
The high-level qualifier name used for the translation table.

mlq
The middle-level qualifier name used for the translation table.

System action

The REXEC client uses the user-specified translation table for all data output.

Operator response

Not applicable.

System programmer response

Not applicable.

User response

Verify that the correct translation table data set is being loaded and used. If the correct table is not being loaded, change the configuration so that the correct translation table data set is available to the REXEC client. See the information about understanding configuration search orders in z/OS Communications Server: IP Configuration Guide for information about searching for translation table data sets.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: REXEC

Module

EZATREXE

Routing code

Not applicable.

Descriptor code

Not applicable.

Example

```
EZA5061I Using User-specified translation table: USER1.FRENCH.TCPXLBIN
```

EZA5575E **NO DESTINATION SPECIFIED FOR NOTE**

Explanation

The user attempted to send a note using the SMTPNOTE EXEC without specifying a destination. Without a destination, the note cannot be delivered.

System action

SMTPNOTE fails.

Operator response

Specify a destination and resend the note.

System programmer response

Assist the user as necessary.

Module

SMTPNOTE

Procedure name

main

EZA5576E **UNABLE TO ALLOCATE TEMPORARY DATASET '*data_set*'**

Explanation

SMTPNOTE is unable to allocate the indicated temporary data set, which is used to hold a current note, because of lack of available storage.

System action

SMTPNOTE fails.

Operator response

Notify the system programmer of the error.

System programmer response

Increase the region size and restart SMTPNOTE.

Module

SMTPNOTE

Procedure name

main.

EZA5577I **NOTE CANCELLED**

Explanation

A note created using SMTPNOTE has been canceled by the user.

System action

SMTPNOTE fails.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5578I **UNABLE TO EDIT TEMPORARY DATASET ‘*data_set*’**

Explanation

SMTPNOTE is unable to edit the indicated temporary data set, which is used to hold a note being created. This message will be accompanied by messages from the operating system giving further information about the reason.

System action

SMTPNOTE fails.

Operator response

Notify the system programmer of this message.

System programmer response

Increase the region size and restart SMTPNOTE.

Module

SMTPNOTE

Procedure name

main

EZA5579E **UNABLE TO TRANSMIT DATA TO ‘*name*’ AT ‘*node*’**

Explanation

SMTPNOTE is unable to transmit a note to the indicated user at the indicated address.

name is the name of the CSSMTP external writer name. For SMTPNOTE CLIST the customization variable is SMTPJOB whose value is used for *name*.

node is the NJE node on which the CSSMTP application is running. For SMTPNOTE CLIST the customization variable is SMTPNODE whose value is used for *node*.

System action

SMTPNOTE fails.

Operator response

Notify the system programmer of the error.

System programmer response

Check that the name is correct for the CSSMTP external writer name. Check that the node is correct NJE node name where the CSSMTP application is running. Check that the node exists in the data set IKJTSOxx for the TRANSREC parameter (NODESMF). Make any changes to the SMTPNOTE customization variables or IKJTSOxx data set and try SMTPNOTE again. For more information about the customization of SMTPNOTE see z/OS Communications Server: IP Configuration Guide.

Module

SMTPNOTE

Procedure name

main

EZA5580E **UNABLE TO DELETE TEMPORARY DATASET '*data_set*'**

Explanation

SMTPNOTE is unable to delete the temporary data set used to hold notes being created. This message will be accompanied by messages from the operating system indicating the reason for the error.

System action

SMTPNOTE fails.

Operator response

Notify the system programmer of the error.

System programmer response

Use the messages displayed by the operating system to determine the cause of the error, and respond as indicated.

Module

SMTPNOTE

Procedure name

main

EZA5581E **CONFLICTING PARAMETERS 'DATASET' AND 'REUSE'**

Explanation

The REUSE parameter, which is used to reuse an existing temporary data set to hold notes being created by SMTPNOTE, was specified, but no temporary data set currently exists. The REUSE parameter is ignored and a new temporary data set is created.

System action

SMTPNOTE fails.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5582E **UNABLE TO USE DATASET '*data_set*'**

Explanation

SMTPNOTE is unable to access the indicated data set. This message is accompanied by messages from the operating system indicating the reason for the error.

System action

SMTPNOTE fails.

Operator response

Verify that the data set is protected. Use the DATESET parameter passed to SMTPNOTE to specify a different data set to be used. If that is not the problem, notify the system programmer of the error.

System programmer response

SMTPNOTE uses the TSO allocation (alloc) command to create a temporary data set similar to the data set passed on the DATESET parameter. The allocation command is failing. Use the messages issued by the operating system to determine the cause of the error and respond as indicated. For more information, see the z/OS TSO/E Command Reference.

Module

SMTPNOTE

Procedure name

main

EZA5583E **UNABLE TO USE DATASET '*data_set*', COPYING ERRORS DETECTED**

Explanation

SMTPNOTE cannot use the indicated data set because it is a protected data set.

System action

SMTPNOTE fails.

Operator response

Verify that the data set logical record (LRECL) attribute is 243 or less. Also, verify that the data set is not being shared with other users when SMTPNOTE is executed. If this is not the problem, notify the system programmer of the error.

System programmer response

SMTPNOTE used the TSO REPRO command to copy the user's data into a temporary data set. This command is failing. Use the messages issued by the operating system to determine the cause of the error and respond as indicated. For more information, see the z/OS DFSMS Access Method Services Commands.

Module

SMTPNOTE

Procedure name

main

EZA5584E **INVALID PARAMETER, *parameter***

Explanation

SMTPNOTE has received an incorrect parameter. The only valid parameters for SMTPNOTE are BATCH, DEBUG, and REUSE.

System action

SMTPNOTE fails.

Operator response

Specify a valid parameter for SMTPNOTE. For more information, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5585E **BATCH PARAMETER REQUIRES DATASET PARAMETER**

Explanation

The BATCH parameter was submitted without a data set name. The BATCH parameter requires a data set name to be imported into the note.

System action

SMTPNOTE fails.

Operator response

Resubmit the SMTPNOTE command, specifying a valid data set to be imported into the note using the BATCH parameter. For more information, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5586E CONFLICTING PARAMETERS ‘CC’ AND ‘NOCC’

Explanation

Both the CC parameter, which specifies addresses to be copied on a note, and the NOCC parameter, which specifies that no addresses are to be copied on the note, were specified. These parameters are mutually exclusive.

System action

SMTPNOTE fails.

Operator response

Submit only one of these parameters when using SMTPNOTE.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5587E NO VALUE FOR PARAMETER ‘TO’

Explanation

No address or user ID was specified in the TO field of the note being created by SMTPNOTE. Without a note or user ID in this field, the note cannot be sent.

System action

SMTPNOTE fails.

Operator response

Specify an address or user ID to receive the note in the TO field of the note. For more information, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5588E NO VALUE FOR PARAMETER ‘CC’

Explanation

The CC parameter was specified, indicating that one or more addresses are to be copied on the current note, but no addresses or user IDs were specified. Without an address or user ID, no copies of the note can be sent.

System action

SMTPNOTE fails.

Operator response

Resubmit the SMTPNOTE command, supplying an address or user ID to receive a copy of the note using the CC parameter. For more information, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main

EZA5589E NO VALUE FOR PARAMETER ‘DATASET’

Explanation

The DATASET parameter, which indicates that a data set is to be imported into the note, was specified without a data set name. Without a data set name, no data set can be imported.

System action

SMTPNOTE fails.

Operator response

Resubmit the SMTPNOTE command, specifying a data set to be included in the note using the DATASET parameter. For more information, see the z/OS Communications Server: IP User's Guide and Commands.

System programmer response

No action is needed.

Module

SMTPNOTE

Procedure name

main.

EZA5590S **UNKNOWN ERROR CODE '*errcode*'**

Explanation

SMTPNOTE has encountered an error that it does not recognize.

System action

SMTPNOTE fails.

Operator response

Notify the system programmer of the error.

System programmer response

Obtain a trace log for the SMTP host and use this information to determine the cause of the error. If no error is apparent, contact the IBM Software Support Center for assistance.

Module

SMTPNOTE

Procedure name

main

EZA5775I **Usage: SNALINK tcpipname applid**

Explanation

SNALINK was invoked with incorrect command line parameters. Correct command line syntax is shown.

System action

SNALINK exits.

Operator response

Try the command again with the correct parameters.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

Main

EZA5776E **Invalid max RU size code**

Explanation

An optional third argument to SNALINK is the maximum RU size code. This parameter was specified on the command line, but is not valid. See z/OS Communications Server: SNA Programming for more information.

System action

SNALINK exits.

Operator response

Try the command again with the correct parameters.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

Main

EZA5777E **Can not obtain storage for path index table**

Explanation

SNALINK could not allocate storage for the path index table.

System action

SNALINK continues to attempt opening the ACB until the open succeeds or it is stopped.

Operator response

No action is needed.

System programmer response

Check that the SNALINK address space is defined correctly.

Module

SNALINK

Procedure name

Main

EZA5778E **Error *error* in DLC INIT call**

Explanation

SNALINK encountered an error issuing a DLC INIT request to initialize DLC.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

SNALINK

Procedure name

Main

EZA5779E **Error *error* IN VTAM OPEN**

Explanation

SNALINK encountered an error issuing a VTAM OPEN request.

System action

SNALINK abends.

Operator response

Check that VTAM is operational.

System programmer response

Check the SNALINK parameters for the correct VTAM_APPL_ID. Contact your VTAM administrator.

Module

SNALINK

Procedure name

Main

EZA5780I **Init complete, APPLID *id*, TCPIP id**

Explanation

SNALINK is running.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

pendconn

EZA5781I **Maximum RU size is *value***

Explanation

This message displays the hexadecimal value of the maximum RU size.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

Main

EZA5782S **TPEND exit called with reason *value***

Explanation

SNALINK exits with the indicated reason code.

System action

SNALINK exits.

Operator response

No action is needed.

System programmer response

Correct the error indicated by the reason code.

Module

SNALINK

Procedure name

Main

EZA5783I ***date time {luname}*** **| Received STOP command or HALT sub-command, shutting down**

Explanation

The operator has entered a MODIFY command specifying HALT or a STOP command. SNALINK begins to shut down normally.

System action

SNALINK terminates.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK ASSEMBLE

Procedure name

PROCCONS/PROCMODF

EZA5784E **Can not free path index table storage**

Explanation

SNALINK encountered an error while freeing storage.

System action

SNALINK exits.

Operator response

No action is needed.

System programmer response

Check the MVS system for system problems.

Module

SNALINK

Procedure name

Main

EZA5785E **Unexpected iptype *type* received**

Explanation

SNALINK received an IUCV interrupt with an unexpected or out-of-range type.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a probable TCPIP code problem. Contact the IBM Software Support Center.

Module

SNALINK

Procedure name

prociucv

EZA5787E **Can't find CONNBLOK for severed path *path***

Explanation

SNALINK received a SEVER interrupt from TCPIP, but could not find the associated connection in its tables. The connection is severed.

System action

SNALINK continues processing.

Operator response

No action is needed.

System programmer response

This message indicates a probable TCPIP code problem. Contact the IBM Software Support Center.

Module

SNALINK

Procedure name

sever

EZA5788E **PENDCONN received from wrong TCPIP id *id***

Explanation

SNALINK received a Connection Pending interrupt from a TCPIP ID other than the one specified on the command line. The pending connection is severed.

System action

SNALINK continues processing.

Operator response

No action is needed.

System programmer response

Check TCPIP parameters (MVS) for the correct TCPIP data set.

Module

SNALINK

Procedure name

pendconn

EZA5789E **Cannot connect to LU *name* — No more CONNBLOKs!**

Explanation

SNALINK does not have enough data structures (CONNBLOKs) allocated to service a pending connection. The pending connection is severed. If existing connections are later closed, resources are freed to allow SNALINK to service this connection.

System action

SNALINK continues processing.

Operator response

No action is needed.

System programmer response

Increase the maximum connections allowed.

Module

SNALINK

Procedure name

pendconn

EZA5790S **Severing pending conn because existing session not killed**

Explanation

SNALINK received a Connection Pending interrupt for this LU. However, a connection already exists for this LU. The pending connection is severed.

System action

SNALINK continues processing.

Operator response

No action is necessary if the connection you want is up; otherwise, stop and restart SNALINK.

System programmer response

This error indicates that TCPIP is out of sync or that there is a code problem.

Module

SNALINK

Procedure name

pendconn

EZA5795E **Unexpected GENTYPE *value* found**

Explanation

SNALINK encountered an incorrect GENTYPE value when processing a VTAM exit.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible TCPIP code problem. Contact the IBM Software Support Center.

Module

SNALINK

Procedure name

procvtam

EZA5796E **Unexpected control field *value* IN SCIP exit**

Explanation

SNALINK encountered an incorrect control field in a SCIP information block, while SNALINK was either acting as an SLU for a connection and waiting for the PLU to establish the connection, or receiving an UNBIND request.

System action

SNALINK ignores the SCIP exit and resumes processing.

Operator response

Tell your VTAM administrator.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamscip

EZA5797E **Rejecting bind from *plu* - no DLC found**

Explanation

SNALINK received a BIND request from the indicated primary LU. It is rejecting the BIND request because it cannot find an associated pending DLC connection in its tables.

System action

The BIND request is rejected.

Operator response

Use the NETSTAT command to check that TCPIP started SNALINK.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

scipbind

EZA5798E **Rejecting bind - already have receive session**

Explanation

SNALINK received a BIND request for an LU on which it has already established communication.

System action

The BIND request is rejected.

Operator response

Tell your VTAM administrator.

System programmer response

This error indicates a possible VTAM problem.

Module

SNALINK

Procedure name

scipbind

EZA5799I **Received BIND request for SNA receive session**

Explanation

A BIND request has been received.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

scipfoun

EZA5800E **Error in MODCB NIB for receive session. R15 *value* R0 *value***

Explanation

SNALINK encountered an error modifying a network information block in preparation for an OPNSEC call to accept a BIND for this LU.

System action

The BIND request is rejected.

Operator response

Tell your VTAM administrator.

System programmer response

This error indicates a possible TCPIP code problem. Contact the IBM Software Support Center.

Module

SNALINK

Procedure name

scipfoun

EZA5801I **Got UNBIND for receive session**

Explanation

An UNBIND request has been received.

System action

The session is brought down.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

scipunbd

EZA5802I **Got UNBIND for send session**

Explanation

An UNBIND request has been received.

System action

The session is brought down.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

scipunbd

EZA5803E **Zero user word in SCIP exit for UNBIND**

Explanation

SNALINK encountered a user word containing zero in an UNBIND request, indicating a null connection block address.

System action

The UNBIND request is not processed.

Operator response

No action is needed.

System programmer response

This error indicates a possible TCPIP or VTAM code problem. Tell your VTAM administrator.

Module

SNALINK

Procedure name

scipunbd

EZA5804E **CID in SCIP UNBIND parm list not found**

Explanation

SNALINK received an UNBIND request that was not associated with either the send or receive session on its LU.

System action

The UNBIND request is not processed.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Module

SNALINK

Procedure name

scipunbd

EZA5805I **TESTCB shows send session not established**

Explanation

SNALINK received a VTAM OPNDST EXIT interrupt specifying a session that has not been established.

System action

SNALINK ends the connection.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem.

Module

SNALINK

Procedure name

vtamopnd

EZA5806E **OPNDST exit can’t find CONNBLOK. RPL addr *address***

Explanation

SNALINK received a VTAM OPNDST EXIT interrupt for which it has no associated connection block.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem.

Module

SNALINK

Procedure name

vtamopnd

EZA5807I **SNA send session established**

Explanation

An SNA send session was established.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamopnd

EZA5808I **TESTCB shows receives session not established**

Explanation

SNALINK received a VTAM OPNSEC EXIT interrupt specifying a session that has not been established.

System action

SNALINK ends the connection.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem. Tell your VTAM administrator.

Module

SNALINK

Procedure name

vtamopns

EZA5809E **OPNSEC exit can’t find CONNBLOK. RPL addr *address***

Explanation

SNALINK received a VTAM OPNSEC EXIT interrupt for which it has no corresponding connection block. SNALINK abends.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem.

Module

SNALINK

Procedure name

vtamopns

EZA5810I **SNA receive session established**

Explanation

An SNA receive session was established.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamopnd

EZA5811E **SEND exit finds zero RPLUSFLD**

Explanation

SNALINK received a VTAM SEND EXIT interrupt with a null connection block pointer.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP error.

Module

SNALINK

Procedure name

vtamsend

EZA5812E **RECEIVE-ANY shows RESPOND *value* CONTROL *control* RH *value***

Explanation

SNALINK handled a RECEIVE-ANY interrupt with the indicated fields.

System action

SNALINK ends the connection, if one exists. Otherwise, the interrupt is ignored.

Operator response

No action is needed.

System programmer response

Tell your VTAM administrator.

Module

SNALINK

Procedure name

rcvisany

EZA5813E **Chaining err: Flags *flag* Buffer *buffer* Pointer *pointer***

Explanation

SNALINK encountered an inconsistency in its buffer chain.

System action

SNALINK ends the connection.

Operator response

No action is needed.

System programmer response

This error indicates a possible TCPIP code problem.

Module

SNALINK

Procedure name

rcvchker

EZA5814E **RECEIVE exit finds zero RPLUSFLD**

Explanation

SNALINK encountered a VTAM RECEIVE EXIT interrupt with a null connection block pointer.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP problem.

Module

SNALINK

Procedure name

vtamrcv

EZA5815I **RECEIVE-ANY shows RESPOND *value* CONTROL *control* RH *value***

Explanation

SNALINK handled a RECEIVE-ANY interrupt with the indicated fields.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamrcv

EZA5816E **CLSDST exit can't find CONNBLOK. RPL addr *address***

Explanation

SNALINK received a VTAM CLSDST EXIT interrupt with an incorrect connection block pointer.

System action

SNALINK abends.

Operator response

Restart SNALINK.

System programmer response

This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Module

SNALINK

Procedure name

vtamclsd

EZA5817E **TERMSESS exit can’t find CONNBLOK. RPL addr *address***

Explanation

SNALINK received a VTAM TERMSESS EXIT interrupt with an incorrect connection block pointer.

System action

SNALINK abends.

Operator response

Tell your VTAM administrator.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamterm

EZA5818I **LOSTERM reason *rc* for receive session**

Explanation

SNALINK received a VTAM LOSTERM EXIT interrupt, which indicates that this LU lost its receive session for the indicated reason code.

System action

SNALINK ends the connection.

Operator response

Check your hardware or VTAM for errors.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlost

EZA5819I **LOSTERM reason *rc* for send session**

Explanation

SNALINK received a VTAM LOSTERM EXIT interrupt, which indicates that this LU lost its receive session for the indicated reason code.

System action

SNALINK ends the connection.

Operator response

Check your hardware or VTAM for errors.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlost

EZA5820I **Zero CONNBLOK ptr in LOSTERM exit, reason *rc***

Explanation

SNALINK received a VTAM LOSTERM EXIT interrupt with a null connection block pointer.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP problem. Tell your VTAM administrator.

Module

SNALINK

Procedure name

vtamlost

EZA5821I **CID in LOSTERM parm list not found, reason *rc***

Explanation

SNALINK received a VTAM LOSTERM EXIT interrupt for which it has no parameter list.

System action

SNALINK ignores the interrupt.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Module

SNALINK

Procedure name

vtamlost

EZA5822I **NSEXIT CLEANUP request for receive session**

Explanation

SNALINK received a VTAM NSEXIT interrupt with a cleanup request for this LU's receive session.

System action

SNALINK ends the connection.

Operator response

Check your hardware or VTAM for errors.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamns

EZA5823I **NSEXIT CLEANUP request for send session**

Explanation

SNALINK received a VTAM NSEXIT interrupt with a cleanup request for this LU's send session.

System action

SNALINK ends the connection.

Operator response

Check your hardware or VTAM for errors.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamns

EZA5824E **Zero CONNBLOK ptr in NSEXIT exit**

Explanation

SNALINK received a VTAM NSEXIT interrupt with a null connection block pointer.

System action

SNALINK abends.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP problem. Tell your VTAM administrator.

Module

SNALINK

Procedure name

vtamns

EZA5825E **CID in NSEXIT parm list not found**

Explanation

SNALINK received a VTAM NSEXIT interrupt for which an associated parameter list could not be found.

System action

SNALINK ignores the interrupt.

Operator response

No action is needed.

System programmer response

This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Module

SNALINK

Procedure name

vtamns

EZA5826I **Sending BIND request for SNA send session**

Explanation

SNALINK is trying to establish the connection. The other side should have a DLC path pending.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

dovtopnd

EZA5827E **Error in MODCB NIB for send session. R15 *value* R0 *value***

Explanation

SNALINK received a VTAM OPNDST interrupt but encountered an error when processing the OPNDST. An error indication is returned to the caller.

System action

The BIND request is rejected.

Operator response

Tell your VTAM administrator.

System programmer response

This error indicates a possible TCPIP code problem.

Module

SNALINK

Procedure name

dovtopnd

EZA5828I **Cannot issue OPNDST - Previous OPNDST active**

Explanation

SNALINK encountered an error processing a VTAM OPNDST request. The associated request parameter list (RPL) is in use, indicating a previous OPNDST is active.

System action

SNALINK does not process the VTAM OPNDST request.

Operator response

No action is needed.

System programmer response

This error indicates a possible TCPIP code problem.

Module

SNALINK

Procedure name

dovtopnd

EZA5829I **Accepting DLC path**

Explanation

A BIND was received. SNALINK accepts a DLC connection to TCPIP.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

doiucacc

EZA5830E **Error *error* accepting DLC path**

Explanation

SNALINK encountered an error in a DLC ACCEPT request.

System action

SNALINK ends the current connection and returns to its caller.

Operator response

No action is needed.

System programmer response

This error indicates a possible TCPIP code problem.

Module

SNALINK

Procedure name

Main

EZA5831E **Error *error* sending on DLC path**

Explanation

SNALINK encountered an error sending data on a DLC path.

System action

SNALINK ends the current connection.

Operator response

No action is needed.

System programmer response

This error indicates a possible system problem.

Module

SNALINK

Procedure name

doiucsnd

EZA5832E **Error *error* receiving on DLC path**

Explanation

SNALINK encountered an error receiving data on a DLC path.

System action

SNALINK ends the current connection.

Operator response

Check that TCPIP is operational.

System programmer response

This error indicates a possible TCPIP code problem.

Module

SNALINK

Procedure name

doiucrcv

EZA5833E **Error *error* severing DLC path**

Explanation

SNALINK encountered an error severing a DLC path.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the *error* portion of this message.

Module

SNALINK

Procedure name

dosever

EZA5834E **Error *error* in VTAM CLOSE**

Explanation

SNALINK encountered an error closing an ACB.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the *error* portion of this message.

Module

SNALINK

Procedure name

closeacb

EZA5835E **Error in GENCB RPL. R15 *value* R0 *value***

Explanation

SNALINK received an error from a GENCB RPL call.

System action

SNALINK resumes processing. More errors can result.

Operator response

No action is needed.

System programmer response

Check the error indication.

Module

SNALINK

Procedure name

setupcon

EZA5836E **Error in GENCB NIB. R15 *value* R0 *value***

Explanation

SNALINK received an error from a GENCB NIB call.

System action

SNALINK resumes processing. More errors can result.

Operator response

No action is needed.

System programmer response

Check the error indication.

Module

SNALINK

Procedure name

setupcon

EZA5837E ***function*** **err. R15** ***value*** **R0** ***value*** **RTNCD** ***rc*** **FDBK2** ***value***

Explanation

SNALINK encountered an error in the indicated function. The associated sense code returned by VTAM is displayed.

System action

Based on the sense code, TCPIP either tries again, ends the session, or ignores the error and continues processing.

Operator response

No action is needed.

System programmer response

Tell your VTAM administrator.

Module

SNALINK

Procedure name

prtrpler

EZA5838E ***function*** **CHECK err. R15** ***value*** **R0** ***value*** **RTNCD** ***rc*** **FDBK2** ***value***

Explanation

SNALINK encountered an error in the indicated function. The associated sense code returned by VTAM is displayed.

System action

Based on the sense code, TCPIP either tries again, ends the session, or ignores the error and continues processing.

Operator response

No action is needed.

System programmer response

Tell your VTAM administrator.

Module

SNALINK

Procedure name

prtchker

EZA5839E ***function*** **sense: SSENSEI, SSENSMU, USENSEI:** ***value date time*** **{*lu*} : SSENSE0,SSENSM0,USENSEO:** ***value***

Explanation

This message lists the sense fields from a SNALINK RPL.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

prtsense

EZA5840E **SSENSEO, SSENSMO, USENSEXO:** ***value***

Explanation

This message displays Sense information.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

prtsense

EZA5841E **ANALYZE returns *value***

Explanation

ANALYZE returned the specified value.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

analyze

EZA5842E **The session type, *type*, invalid. Valid values are: SINGLE, DUAL or blank**

Explanation

The session type that you entered is not valid.

System action

SNALINK abends.

Operator response

Check the SNALINK parameters, and correct them as needed. See z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

chkmod

EZA5843I **Using DUAL LU0 sessions**

Explanation

This mode is compatible with previous versions of SNALINK.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

chkmod

EZA5844I **Using SINGLE LU0 sessions**

Explanation

This mode is not compatible with previous versions of SNALINK.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

chkmod

EZA5845I **Got UNBIND for session**

Explanation

An UNBIND request was received for a single session.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

scipunbd

EZA5846I **Received BIND request for SNA session**

Explanation

A BIND request was received for a single session.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

scipfoun

EZA5847I **SNA session established**

Explanation

A single session was established.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamopnd

EZA5848E **LOSTERM received, reason *rc***

Explanation

A LOSTERM request was received.

System action

SNALINK ends the connection.

Operator response

Check your hardware or VTAM for problems.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlost

EZA5849E **NSEXIT CLEANUP request for session**

Explanation

A VTAM NSEXIT interrupt with cleanup request was received.

System action

SNALINK ends the connection.

Operator response

Check your hardware or VTAM for problems.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamns

EZA5850I **Sending SNA BIND request**

Explanation

A connection is being established.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlgon

EZA5851E **Rejecting logon attempt from *id* -no DLC found**

Explanation

There is no corresponding DLC connection to TCPIP for the indicated ID. No connection is established.

System action

Processing continues.

Operator response

Check that the SNALINK device is running.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlgon

EZA5852E **Rejecting logon attempt - already have a session with *id***

Explanation

A BIND request was received for an LU that has already established a connection.

System action

The BIND is not accepted.

Operator response

Tell your VTAM administrator.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlgon

EZA5853I **Processing LOGON exit**

Explanation

The requested logon is being processed.

System action

A BIND request is sent.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

vtamlgon

EZA5854E **Invalid number of sessions, using default of 6 sessions**

Explanation

The maximum session parameter is not valid.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

To avoid this error, correct the maximum session parameter.

Module

SNALINK

Procedure name

main

EZA5855E **Invalid delay interval, using default of 15 minutes**

Explanation

The retry interval parameter is not valid.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

To avoid this error, correct the retry interval parameter.

Module

SNALINK

Procedure name

main

EZA5856I **REPLY X TO SHUT DOWN**

Explanation

SNALINK is shutting down.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

main

EZA5864I **Max Sessions *value* : Retry Interval *value***

Explanation

This message displays the values set for the maximum number of sessions and the retry interval being used by SNALINK.

System action

Processing continues.

Operator response

Correct the values if necessary.

System programmer response

No action is needed.

Module

SNALINK

Procedure name

main

EZA5866W ***date time {luname}* | Command is too long, truncating**

Explanation

The operator has entered a MODIFY command that is longer than 254 characters. If the first 254 characters of the command are syntactically correct then the command is executed with no additional messages. Otherwise, additional messages will be generated.

System action

Processing continues.

Operator response

The command should be reentered as several shorter commands.

System programmer response

No action is needed.

Module

SNALINK ASSEMBLE

Procedure name

PROCMOD

EZA5867E ***date time {luname}* | Invalid MODIFY sub-command.**

Explanation

The operator has entered a MODIFY command with a subcommand which was neither HALT nor PKTTRACE. The command is ignored.

System action

Processing continues.

Operator response

Specify the MODIFY command with either the HALT or PKTTRACE subcommand.

System programmer response

No action is needed.

Module

SNALINK ASSEMBLE

Procedure name

PROCMODF

EZA5868W ***date time {luname}* | START CIB not freed, MODIFY Commands restricted**

Explanation

An error occurred during the initialization of MODIFY command processing. The command input buffer (CIB) containing the START command was not freed from the task CIB chain. This is an internal error. It will not cause a system problem.

System action

SNALINK continues.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center to report this error. MODIFY commands may result in IEE342I messages from the MVS operating system. See z/OS MVS System Messages, Vol 7 (IEB-IEE) for an explanation of the IEE342I message.

Module

SNALINK ASSEMBLE

Procedure name

MAIN

EZA5869E ***date time {luname}* | PKTTRACE Disabled, TCP/IP links not connected**

Explanation

The operator has attempted to issue a PKTTRACE subcommand before TCPIP has connected to SNALINK.

SNALINK receives link information from TCPIP before establishing VTAM sessions. Packet tracing is enabled when link information has been received from TCPIP. The PKTTRACE subcommand is ignored.

System action

Processing continues.

Operator response

Wait for the IUCV connection between TCPIP and SNALINK to be established and reenter the command.

System programmer response

No action is needed.

Module

SNALINK ASSEMBLE

Procedure name

PROCMODF

EZA5926I **LU62CFG: Starting Pass *value* of *value***

Explanation

The indicated pass number of the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG has just been started. If errors are not detected in earlier passes, the data set is parsed by way of the number of indicated passes.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62INIT

Procedure name

RLCFG#P1, RLCFG#P2

EZA5927I **LU62CFG: No Errors Detected - Initialization will Continue**

Explanation

No errors were detected in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

Initialization continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62INIT

Procedure name

L62ENTRY

EZA5928I **LU62CFG: Errors Detected - Address Space will Terminate**

Explanation

One or more errors were detected during the parsing of the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

The address space shuts down immediately.

Operator response

No action is needed.

System programmer response

See the previous console messages and messages written to the SYSPRINT data set to determine the exact nature of the errors. Correct the errors, and restart the SNALINK LU6.2 interface.

Module

LU62INIT

Procedure name

L62ENTRY

EZA5929I **Address Space *address* Already Active - this Address Space will Terminate**

Explanation

The indicated address space is already running; each instance of a SNALINK LU6.2 interface should have a unique address space name.

System action

This address space shuts down immediately.

Operator response

No action is needed.

System programmer response

If multiple SNALINK LU6.2 address spaces are required, each one should have a unique name.

Module

LU62INIT

Procedure name

L62ENTRY

EZA5930I **Errors Detected in VTAM Initialization - Address Space will Terminate**

Explanation

One or more errors were detected during the initialization of the interface with the VTAM subsystem.

System action

The address space shuts down immediately.

Operator response

No action is needed.

System programmer response

See the previous console messages and messages written to the SYSPRINT data set to determine the exact nature of the errors. If possible, correct the errors, and restart the SNALINK LU6.2 interface. Otherwise, if the previous error messages indicate a potential program logic error, contact the IBM Software Support Center.

Module

LU62INIT

Procedure name

L62ENTRY

EZA5931I **Errors Detected in DLC Initialization - Address Space will Terminate**

Explanation

One or more errors were detected during the initialization of the DLC interface between the SNALINK LU6.2 address space and the TCPIP address space.

System action

The address space shuts down immediately.

Operator response

No action is needed.

System programmer response

See the previous console messages and messages written to the SYSPRINT data set to determine the exact nature of the errors. If possible, correct the errors, and restart the SNALINK LU6.2 interface. Otherwise, if the previous error messages indicate a potential program logic error, contact the IBM Software Support Center.

Module

LU62INIT

Procedure name

L62ENTRY

EZA5932I **Initialization complete - Applid: *application* TCP/IP: *address***

Explanation

The initialization of the SNALINK LU6.2 interface has been successfully completed. The indicated application ID is used for VTAM-related communication, and DLC connections are only accepted from the indicated address space.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMINIT

EZA5933I **Link *name* opened**

Explanation

A DLC connection associated with the indicated TCPIP link has been successfully opened. Datagrams to and from destinations associated with the indicated link are passed between the TCPIP address space and the SNALINK LU6.2 address space using the DLC connection.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62IUCV

Procedure name

IUCVPCON

EZA5934I **Establishing connection with *name* for *address***

Explanation

An SNA LU type 6.2 connection has been initiated with the indicated remote LU name. This connection is used as the medium through which data to and from the indicated IP address are transferred.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA5935I **Send conversation allocated for *address***

Explanation

The SNA LU type 6.2 conversation that is used for sending data to the destination node associated with the indicated IP address has been successfully established. IP datagrams addressed to the indicated address are sent to the destination node using the send conversation.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA5936I **Receive conversation allocated for *address***

Explanation

The SNA LU type 6.2 conversation to be used for receiving data from the destination node associated with the indicated IP address has been successfully established. IP datagrams from the indicated IP address are transferred to the local node using the receive conversation.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA5937I **Link *name* closed**

Explanation

The DLC connection associated with the indicated TCPIP link has been closed. The SNALINK LU6.2 interface attempts to buffer any datagrams received from destination nodes associated with the indicated link. However, if the link is not reestablished before the buffer space is exhausted, incoming datagrams are discarded. The TCPIP address space is responsible for the reconnection of the DLC link.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

If the SNALINK LU6.2 address space is not shut down, check the reason for the severing of the DLC connection. The TCPIP address space might have been shut down or abended.

Module

LU62IUCV

Procedure name

IUCVSEVR, IUCVCMES

EZA5938I **Received operator shutdown request**

Explanation

A console command requesting the orderly shutdown of the SNALINK LU6.2 interface has been received and is processed immediately.

System action

The SNALINK LU6.2 address space shuts down immediately.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MAIN

Procedure name

MAINPROC

EZA5939I **TPEND exit called with reason *rc***

Explanation

A VTAM shutdown notification was received resulting in the invocation of the TPEND exit routine. The indicated reason code was passed to the exit routine.

System action

The SNALINK LU6.2 address space shuts down immediately.

Operator response

No action is needed.

System programmer response

Determine the cause of the VTAM shutdown notification. For more information about VTAM shutdown notifications see z/OS Communications Server: SNA Programmer's LU 6.2 Reference.

Module

LU62MAIN

Procedure name

MAINPROC

EZA5940I **TCPIP ADDRESS SPACE NAME SET TO *AddressSpaceName***

Explanation

The TCPIP address space name to be used by SNALINK LU6.2 has been set to *AddressSpaceName*. It corresponds to the value of the TCPIPJOBNAME statement found in TCPIP.DATA.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the indicated name is the correct TCPIP address space name to be used by SNALINK LU6.2. If the name is incorrect, specify the correct name as the value of the TCPIP.DATA TCPIPJOBNAME statement and restart the SNALINK LU6.2 application.

Module

LU62INIT

Procedure name

READTCPD

EZA5950W **LU62CFG: Line *number*: DEST statement - Blank Connection Start Type, *type* assumed**

Explanation

The indicated line of the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG contains a DEST statement that does not include a specification of when the connection to the destination is to be established. The indicated start type value is assumed. The indicated start type value is used for the destination defined on the indicated line of the configuration data set.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the indicated start type is the required start type value for the destination. If the value is incorrect, correct the SNALINK LU6.2 configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA5951W **LU62CFG: Line *number*: BUFFERS statement - Blank number of Additional Send Buffers, *value* assumed**

Explanation

The indicated line of the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG contains a BUFFERS statement that does not include a specification of the number of additional send buffers to be allocated for each destination. The indicated default value is assumed. The number of additional send buffers allocated for each destination is the indicated number.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the indicated number is the required number of additional send buffers for each destination. If the number is incorrect, correct the SNALINK LU6.2 configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA5952W **LU62CFG: Line *number*: BUFFERS statement - Blank Send Queue Limit, *value* assumed**

Explanation

The indicated line of the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG contains a BUFFERS statement that does not include a specification of the send queue limit to be enforced for each destination. The indicated default value is assumed.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the indicated number is the required send queue limit to be enforced for each destination. If the number is incorrect, correct the SNALINK LU6.2 configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA5953W **LU62CFG: Line *number*: LINK statement - Blank Idle Timeout, *seconds* assumed**

Explanation

The indicated line of the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG contains a LINK statement that does not include a specification of the idle timeout to be used for all connections associated with the link. The indicated default number of seconds is assumed. Inactive connections for the current link are closed after the indicated number of seconds of inactivity.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the indicated number of seconds is the required idle timeout value for each destination associated with the link. If the number of seconds is incorrect, correct the SNALINK LU6.2 configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA5954W **NO TCPIP.DATA FILE ACCESSED - ADDRESS SPACE NAME TCPIP ASSUMED**

Explanation

The TCPIP address space name to be used by SNALINK LU6.2 has been set to TCPIP. This is the default value. It has been assigned because no TCPIPUSERID or TCPIPJOBNAME TCPIP.DATA statement was found.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

If the name is incorrect, specify the correct name as the value of the TCPIP.DATA TCPIPJOBNAME statement. Also check that the SYSTCPD DD statement for the SNALINK LU6.2 procedure has the correct TCPIP.DATA file specified. After making any changes restart the SNALINK LU6.2 application.

Module

LU62INIT

Procedure name

READTCPD

EZA5959I **CANCEL Command Accepted - Address Space will Abend**

Explanation

A CANCEL subcommand was entered from an operator console (using the MODIFY console command).

System action

The SNALINK LU6.2 address space abends immediately. If the SYSUDUMP ddname is defined, a dump is produced.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

MODIFY

EZA5960I **HALT Command Accepted - Address Space will Shutdown**

Explanation

A HALT subcommand was entered from an operator console (using the MODIFY console command).

System action

The SNALINK LU6.2 shuts down immediately.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

MODIFY

EZA5961I **Establishment of Connection to IP Address *address* Initiated**

Explanation

The establishment of a connection to the destination node identified by the indicated IP address has been requested by a RESTART subcommand entered from an operator console (using the MODIFY console command).

System action

A connection to the indicated IP address is initiated.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

STRTCONN

EZA5962I **Drop of Connection to IP Address *address* Initiated**

Explanation

The termination of a connection with the destination node identified by the indicated IP address has been requested by a DROP subcommand entered from an operator console (using the MODIFY console command).

System action

The connection with the indicated IP address is closed.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROPCONN

EZA5963I **TRACE Command Completed**

Explanation

The trace level specified in a TRACE subcommand entered from an operator console (using the MODIFY console command) has been set for the specified destinations. The new trace levels take effect immediately.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA5964I ***address* (Disconnected)**

Explanation

This message is output on the operator console in response to a LIST subcommand (entered using the MODIFY console command). It indicates that no connection with the indicated IP address has been established during the current invocation of the SNALINK LU6.2 address space.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5965I ***address* (Disconnected on *date* at *time*)**

Explanation

This message is output on the operator console in response to a LIST subcommand (entered using the MODIFY console command). It indicates that no connection with the indicated IP address currently exists, but a connection was disconnected at the indicated time on the indicated date. The date is in Julian date format (yy.ddd), and the time is in standard 24-hour clock format (hh:mm:ss).

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5966I ***address* (Connected on *date* at *time*)**

Explanation

This message is output on the operator console in response to a LIST subcommand (entered using the MODIFY console command). It indicates that a connection with the indicated IP address currently exists; it was established at the indicated time on the indicated date. The date is in Julian date format (yy.ddd), and the time is in standard 24-hour clock format (hh:mm:ss). Further details of the status of this connection are output on the console (using a multiline WTO).

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5967I **Connected via: *type* Trace Level: *level***

Explanation

This message is output on the operator console following message EZA5966I to provide further details of the connection identified in the EZA5966I message.

The indicated type describes why the connection was established:

- Data. Connection established either because a datagram was sent or received to or from the destination node, or an INIT option was processed by the destination node;
- Init. Connection established because of the INIT option in the configuration data set;
- Restart. Connection established because of the RESTART subcommand entered on console (using MODIFY console command).

The indicated trace level indicates the current level of tracing in force for the connection ("On," "Off," or "Detail"). Further details of the status of this connection are output on the console (using a multiline WTO).

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5968I **SEND:- Status: *status* Packets Out: *value***

Explanation

This message is output on the operator console following messages EZA5966I and EZA5967I to provide further details of the connection identified in the EZA5966I message.

The indicated status value describes the current status of the send conversation with the conversation node:

- Allocated. Conversation is allocated and ready to send data.
- Not Allocated. Conversation is either yet to be started, or has just been deallocated.
- Pending Alloc. Allocation of conversation has been initiated but not yet completed.
- Pending Dealloc. Deallocation of the conversation has been initiated but not yet completed.

The indicated value describes the number of datagrams sent to the destination node during the period that the current connection has existed. Further details of the status of this connection are output on the console (using a multiline WTO).

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5969I **RECV:- Status: *status* Packets In: *value***

Explanation

This message is output on the operator console following messages EZA5966I, EZA5967I and EZA5968I to provide further details of the connection identified in the EZA5966I message.

The indicated status describes the current status of the receive conversation with the conversation node:

- Allocated. Conversation is allocated and ready to receive data.
- Not Allocated. Conversation is either yet to be started, or has just been deallocated.
- Pending Alloc. Allocation of conversation has been initiated but not yet completed.
- Pending Dealloc. Deallocation of conversation has been initiated but not yet completed.

The indicated value describes the number of datagrams received from the destination node during the period that the current connection has existed. This is the last line in the multiline message for a connected destination.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5970I **LIST Accepted; Range = Single Connection**

Explanation

A LIST subcommand requesting details of the connection status for a single destination has been entered from an operator console (using the MODIFY console command). Subsequent messages are written to the operator console showing the connection status for the specified destination.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5971I **LIST Accepted; Range = All Connections**

Explanation

A LIST subcommand requesting details of the connection status for all destinations has been entered from an operator console (using the MODIFY console command). Subsequent messages are written to the operator console showing the connection status for each of the destinations defined in the SNALINK LU6.2 configuration data set.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5972I **LIST Accepted; Range = All Active Connections**

Explanation

A LIST subcommand requesting details of the connection status for all currently connected destinations has been entered from an operator console (using the MODIFY console command). Subsequent messages are written to the operator console showing the connection status for each of the destinations defined in the SNALINK LU6.2 configuration data set that have currently established connections.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5973I **LIST Completed**

Explanation

The connection status information requested using the LIST subcommand has been written to the operator console.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LISTCONN

EZA5984I **OLU=** ***name*****, DLU=** ***name*****, IP address=** ***address***

Explanation

This message is written to the SYSPRINT data set when tracing is enabled to identify the source and destination LU names relating to the action in the preceding message. It can appear following either EZA5986I, EZA5987I, EZA5988I, or EZA5989I. The indicated OLU name is the originating LU name of the allocation, deallocation, send, or receive identified by the preceding message. The indicated DLU name is the destination LU name.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMALOC, VTAMDALC, VTAMSEND, VTAMRECV

EZA5985I **LU= *name*, Linkname= *name*, IP address = *address***

Explanation

This message is written to the SYSPRINT data set when tracing is enabled to identify the LU name and link name relating to the action in the preceding message. It can appear following either EZA5990I, EZA5991I, EZA5992I, EZA5993I, or EZA5994I. The indicated LU name is the LU name associated with the action described by the preceding message. The indicated link name is the TCPIP link name.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMRECV, ADDVTSND, DOVTALOC

EZA5986I **VTAM conversation allocated; Convid= *id*, SID= *id***

Explanation

This message is written to the SYSPRINT data set when tracing is enabled and a conversation is allocated. The indicated conversation and session IDs are the conversation and session identifiers returned from VTAM. Message EZA5984I appears following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA5987I **VTAM conversation deallocated; Convid= *id*, SID= *id***

Explanation

This message is written to the SYSPRINT data set when tracing is enabled and a conversation is deallocated. The indicated conversation and session IDs are the conversation and session identifiers returned from VTAM. Message EZA5984I appears following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMDALC

EZA5988I **VTAM sent logical record; Convid= *id*, SID= *id*, bytes= *length***

Explanation

This message is written to the SYSPRINT data set when tracing is enabled and a logical record is passed to VTAM for transmission to a destination node. The indicated conversation and session IDs are the conversation and session identifiers returned from VTAM. The indicated length is the length of the VTAM logical record, which includes a (2-byte) GDS ID and a (2-byte) length field as well as the IP datagram. Message EZA5984I is written following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMSEND

EZA5989I **VTAM received logical record; Convid= *id*, SID= *id*, bytes= *length***

Explanation

This message is written to the SYSPRINT data set when tracing is enabled and a logical record was received from VTAM. The indicated conversation and session IDs are the conversation and session identifiers returned from VTAM. The indicated length is the length of the VTAM logical record, which includes a (2-byte) GDS ID and a (2-byte) length field as well as the IP datagram. Message EZA5984I appears following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA5990I **IP datagram packed into message, bytes= *length***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and an IP datagram, which was extracted from a logical record received from VTAM, was loaded into a DLC message buffer. The DLC message buffer is sent to the TCPIP address space (using DLC) once the buffer is filled or there are no more records to receive from VTAM. The indicated length is the length of the IP datagram loaded into the buffer. Message EZA5985I appears following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA5991I **IP datagram added to the pending message queue, bytes= *length***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and an IP datagram, which was extracted from a logical record received from VTAM, was loaded into the VTAM-pending message queue. Datagrams are only loaded into the VTAM-pending message queue if the DLC message buffer is either full or currently being sent to the TCPIP address space. Entries in the queue are placed in the DLC message buffer (and subsequently sent to TCPIP) once the current buffer has been successfully sent to the TCPIP address space. The indicated length is the length of the IP datagram placed on the queue. Message EZA5985I appears following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA5992I **IP datagram added to the VTAM send queue, bytes= *length*, queue count= *number***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and an IP datagram, which was extracted from a DLC message received from TCPIP, was loaded into the VTAM send queue. All datagrams to be sent to destination nodes are placed on the VTAM send queue. From there they are extracted and passed to VTAM for transmission. The indicated number is the number of entries on the VTAM send queue after the new entry has been added. The indicated length is the length of the IP datagram placed on the queue. Message EZA5985I appears following this message.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

ADDVTSND

EZA5993I **CNOS for dependent partner; SESSLIM=*value*, WINNER=*value*, LOSER=*value***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled to show the CNOS limits established with a dependent partner LU. The indicated value for SESSLIM is the maximum number of conversations between the local and destination LUs. The indicated value for WINNER is the number of *contention winner* sessions (the number of send sessions). The indicated value for LOSER is the number of *contention loser* sessions (the number of receive sessions). Message EZA5985I is written to the SYSPRINT data set following this message to identify the connection involved.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

DOVTALOC

EZA5994I **CNOS for independent partner; SESSLIM=*value*, WINNER=*value*, LOSER=*value***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled to show the CNOS limits established with an independent partner LU. The indicated value for SESSLIM is the maximum number of conversations between the local and destination LUs. The indicated value for WINNER is the number of *contention winner* sessions (the number of send sessions). The indicated value for LOSER is the number of *contention loser* sessions (the number of receive sessions). Message EZA5985I is written to the SYSPRINT data set following this message to identify the connection involved.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

DOVTALOC

EZA5995I **Number of IP packets sent on *address* = *value***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and an IP datagram has just been passed to VTAM for transmission to the destination node associated with the indicated IP address. The indicated number is the number of IP datagrams sent to the indicated IP address during the current invocation of the SNALINK LU6.2 address space, including the datagram that triggered this message. One or more lines of message EZA5999I appear following this message to show the contents of the datagram just sent.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMSEND

EZA5996I **Number of IP packets received on *address* = *value***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and an IP datagram has just been received, using VTAM, from the destination node associated with the indicated IP address. The indicated number is the number of IP datagrams received from the indicated IP address during the current invocation of the SNALINK LU6.2 address space, including the datagram that triggered this message.

System action

Processing continues. One or more lines of message EZA5999I appear following this message to show the contents of the datagram just received.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA5997I **Connection *address* will timeout in *seconds* seconds**

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and the inactivity timeout value for the connection associated with the indicated IP address has just been recalculated. The indicated number of seconds is the time until the connection is due to timeout unless the connection is used within that period. If the connection with the indicated IP address is not used within the indicated number of seconds, the connection is closed.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

RESTIMER

EZA5998I **VTAM connection has reached timeout period and expired**

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled and the inactivity timeout period for a connection has expired. Message EZA5985I is written to the SYSPRINT data set following this message to identify the connection involved. The connection identified in message EZA5985I, which follows this message, is closed.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

RESTIMER

EZA5999I ***datagram***

Explanation

This message is written to the SYSPRINT data set when detailed tracing is enabled. It contains a hexadecimal dump of the contents of an IP datagram and is repeated until the entire datagram has been dumped. Each of these messages shows the hexadecimal representation of 40 bytes of data from the datagram. Trailing nulls (X'00') are used to fill the last message for a datagram if required. This message is repeated until the contents of the entire datagram have been written to SYSPRINT.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62VTAM

Procedure name

HEXDUMP

Chapter 8. EZA6*xxxx* messages

EZA6009W **Conversations for *address* terminated**

Explanation

The send and receive conversations comprising the connection to the destination node associated with the indicated IP address have been closed. The connection was closed for one of the following reasons:

- An allocation failure occurred.
- One of the conversations was closed because of a VTAM error or command.
- The connection was closed using a DROP subcommand entered from an operator console (using the MODIFY console command).
- The inactivity timeout period for the connection expired.
- The destination node closed one or both conversations.

Internal data structures are reinitialized to allow the connection to be restarted when required.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

If the termination was caused by an error, see the preceding messages in the SYSPRINT data set to determine the exact nature of the problem.

Module

LU62VTAM

Procedure name

CLEARCON

EZA6010E **Receive error for *address*, what received= *value*, length= *length***

Explanation

An incomplete logical record was received from the destination node associated with the indicated IP address. The indicated value is the **what received** field returned from VTAM. The indicated length is the number of bytes received. This error can be caused by the SNALINK LU6.2 transaction program on the destination node that does not conform to the protocol defined in the relevant RFC.

System action

The connection is closed. Processing continues.

Operator response

No action is needed.

System programmer response

See the z/OS Communications Server: SNA Programmer's LU 6.2 Reference for an explanation of the received value. If possible, correct the error; otherwise, contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA6011E **Unable to allocate send conversation for *address***

Explanation

The send conversation could not be allocated with the destination node associated with the indicated IP address. Preceding messages in the SYSPRINT data set indicate the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See the preceding messages to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA6012E **Unable to flush the FMH5 for *address***

Explanation

The send conversation could not be allocated with the destination node associated with the indicated IP address. The Function Management Header (FMH5) could not be flushed. Messages EZA6029E to EZA6032E are also written to the SYSPRINT data set to describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA6013W **Unable to deallocate conversation for *address***

Explanation

A send and receive conversation with the destination node associated with the indicated IP address could not be deallocated. Messages EZA6029E to EZA6032E are also written to the SYSPRINT data set to describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMDALC

EZA6014W **Rejecting receive conversation allocation attempt for *address***

Explanation

An attempt to allocate a receive conversation with the destination node associated with the indicated IP address was rejected.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See the preceding messages in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6015W **A conversation allocation attempt has been lost**

Explanation

A conversation allocation attempt has been unsuccessful because the Function Management Header (FMH5) was not received correctly. Messages EZA6029E to EZA6032E are also written to the SYSPRINT data set to describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6016W **Not authorized to accept conversation allocation attempt from *lu*.**

Explanation

A conversation allocation attempt was received from the indicated logical unit, but the logical unit is not defined in the SNALINK LU6.2 configuration data set.

System action

Processing continues; the conversation allocation attempt is rejected.

Operator response

No action is needed.

System programmer response

Correct the SNALINK LU6.2 configuration data set and restart the SNALINK LU6.2 address space.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6017W **VTAM event queue full, event lost**

Explanation

The VTAM event queue has overflowed, and an event has been lost.

System action

The SNALINK LU6.2 address space attempts to continue; however, more problems might occur with VTAM interactions.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

PUTVTAM

EZA6018W **Unable to reject conversation for *address***

Explanation

An error was encountered while attempting to reject a receive conversation allocation attempt from the destination node associated with the indicated IP address. Messages EZA6029E to EZA6032E are also written to the SYSPRINT data set to describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6019W **Unable to find destination entry for *lu*; partner for lost conversation**

Explanation

A loss notification was received from VTAM for a conversation with the indicated logical unit; however, the logical unit is not defined in the SNALINK LU6.2 configuration data set. The loss notification is ignored.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Potential problem in either VTAM or SNALINK LU6.2. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6020W **Lost send conversation for *address***

Explanation

A loss notification has been received from VTAM for the send conversation for the connection with the destination node associated with the indicated IP address.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See the VTAM diagnostic messages to determine the reason for the conversation loss.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6021W **Lost receive conversation for *address***

Explanation

A loss notification has been received from VTAM for the receive conversation for the connection with the destination node associated with the indicated IP address.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See the diagnostic messages output by the SNALINK LU6.2 transaction program on the destination node to determine the reason for the conversation loss.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6022W **Closing conversations for *address***

Explanation

The connection to the destination node associated with the indicated IP address is being closed because of an error condition encountered during processing. Preceding error messages in the SYSPRINT data set describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See the preceding messages in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMSEND, DOVTSEND, VTAMRECV, DOVTRECV

EZA6023W **Unable to complete CNOS on LU *lu* for *address***

Explanation

The CNOS macro invocation for the indicated logical unit was unsuccessful. The CNOS macro is the first VTAM macro executed in the conversation allocation process. This error condition occurs when the indicated LU name is either incorrect or unreachable. Messages EZA6029E to EZA6032E are also written to the SYSPRINT data set to describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

DOVTALOC

EZA6024E **DLC event queue full, event lost**

Explanation

The DLC event queue has overflowed, and an event has been lost.

System action

The SNALINK LU6.2 address space abends with 3C5 reason 105.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

PUTIUCV

EZA6025E **Unable to allocate receive conversation for *address***

Explanation

The allocation of the receive conversation with the destination node associated with the indicated IP address was unsuccessful. Preceding messages in the SYSPRINT data set indicate the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See the preceding messages to determine the nature of the problem.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA6026E **Unable to initiate conversation allocation with *lu* for *address***

Explanation

A conversation allocation request for the indicated logical unit was rejected by VTAM. This can be because of incorrect LU or mode names, or corrupted request blocks, control blocks, or both. Messages EZA6029E to EZA6032E are also written to the SYSPRINT data set to describe the exact nature of the problem.

System action

Processing continues; the connection is closed.

Operator response

No action is needed.

System programmer response

See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Module

LU62VTAM

Procedure name

DOVTALOC

EZA6027E **Error *rc* in VTAM OPEN**

Explanation

An error occurred in opening the Access Method Control Block (ACB) defined in the VTAM statement in the SNALINK LU6.2 configuration data set. The indicated return code is the ACB error value returned by VTAM.

System action

The SNALINK LU6.2 address space shuts down immediately.

Operator response

No action is needed.

System programmer response

If the return code = 5A, then verify that the VTAM APPL on the VTAM statement in TCPIPL62.TCPIPL62.CONFIG data set is defined in your VTAM list and has been activated. If other return codes are issued with this message, see z/OS Communications Server: SNA Programmer's LU 6.2 Reference to find the meaning of the indicated ACB error value returned by VTAM.

Module

LU62VTAM

Procedure name

VTAMINIT

EZA6028E **Error *flag* in VTAM CLOSE**

Explanation

An error occurred in closing the Access Method Control Block (ACB) defined in the VTAM statement in the SNALINK LU6.2 configuration data set. The indicated flag is the ACB error flag value returned by VTAM.

System action

The SNALINK LU6.2 address space shuts down immediately.

Operator response

No action is needed.

System programmer response

See the VTAM diagnostic documentation to determine the meaning of the indicated flag.

Module

LU62VTAM

Procedure name

CLOSEACB

EZA6029E ***command* err. R15 *value* R0 *value* RTNCD *rc* FDBK2 *rc***

Explanation

The indicated VTAM APPC command was unsuccessful. The indicated value for R15 is the value returned using general register 15. The indicated value for R0 is the value returned using general register 0. The indicated return code for RTNCD is the return code returned from the APPC command. The indicated return code for FDBK2 is the value of the feedback return code returned using the RPL.

System action

Processing continues; messages EZA6030E to EZA6032E are also written to the SYSPRINT data set following this message to further describe the error condition encountered.

Operator response

No action is needed.

System programmer response

See z/OS Communications Server: SNA Programmer's LU 6.2 Reference for a description of the meanings of each of the values reported in this message.

Module

LU62VTAM

Procedure name

PRTRPXER

EZA6030E ***command* err. RCPRI= *rc*, RCSEC= *rc***

Explanation

The indicated VTAM APPC command was unsuccessful. This message follows message EZA6029E to further describe the error encountered. The indicated return code for RCPRI is the value returned from VTAM using the primary return code field of the RPL. The indicated return code for RCSEC is the secondary return code value. Messages EZA6031E and EZA6032E are also written to the SYSPRINT data set following this message to further describe the error condition encountered.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

See z/OS Communications Server: SNA Programmer's LU 6.2 Reference for a description of the meanings of the primary and secondary return code values reported in this message.

Module

LU62VTAM

Procedure name

PRTRPXER

EZA6031E ***command* err. Sense code received: *value***

Explanation

The indicated VTAM APPC command was unsuccessful. This message follows messages EZA6029E and EZA6030E to further describe the error encountered. The indicated sense code is the sense code value returned from VTAM. Message EZA6032E is also written to the SYSPRINT data set following this message to further describe the error condition encountered.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

See z/OS Communications Server: SNA Programmer's LU 6.2 Reference for a description of the sense code values.

Module

LU62VTAM

Procedure name

PRTSENSE

EZA6032E ***command* err. Sense code specified: *value***

Explanation

The indicated VTAM APPC command was unsuccessful. This message follows messages EZA6029E to EZA6031E to further describe the error encountered. The indicated sense code is the sense code value specified on the VTAM macro invocation that was unsuccessful.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

See z/OS Communications Server: SNA Programmer's LU 6.2 Reference for a description of the sense code values.

Module

LU62VTAM

Procedure name

PRTSENSE

EZA6046E **Rejecting DLC path for *link*, not configured**

Explanation

A DLC path allocation request has been received from the TCPIP address space for the indicated link, but the link is not defined by a LINK statement in the SNALINK LU6.2 configuration data set. The DLC path allocation request is rejected.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Correct the *hlq*.PROFILE.TCPIP data set or the SNALINK LU6.2 configuration data set, so that the link names used in each data set correspond exactly.

Module

LU62IUCV

Procedure name

IUCVPCON

EZA6047E **Rejecting DLC path for *link*, bad entry setup**

Explanation

A DLC path allocation request has been received from the TCPIP address space for the indicated link, but an error was encountered in creating the corresponding DLC table entry. The DLC path allocation request is rejected. The error detected was either that the SNALINK LU6.2 address space has insufficient free storage to allocate the buffers required for the link to be established, or an active entry already exists in the DLC table data structure. A preceding message in the SYSPRINT data set identifies the exact nature of the error.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

If insufficient storage caused the error, increase the REGION size for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62IUCV

Procedure name

IUCVPCON

EZA6048E **Error *rc* accepting DLC path**

Explanation

An error was encountered during the establishment of a DLC path between the TCPIP address space and the SNALINK LU6.2 address space. The DLC path allocation request is rejected. The indicated return code is the code returned using register 15 from the accept attempt.

System action

Processing continues.

Operator response

Check that the TCPIP address space is active.

System programmer response

Check the TCPIP log data set for an indication of the nature of the error. Attempt to try the path establishment again by restarting the SNALINK LU6.2 device from within TCPIP. If the error persists, contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

DOIUACCP

EZA6049E **Error *rc* sending on DLC path**

Explanation

An error was encountered while sending data using a DLC path to the TCPIP address space. The DLC path is severed. The indicated return code is the code returned using register 15 from the send attempt.

System action

Processing continues.

Operator response

Check that the TCPIP address space is active.

System programmer response

Check the TCPIP log data set for an indication of the nature of the error. Attempt to reestablish the IUCV path by restarting the SNALINK LU6.2 device from within TCPIP. If the error persists, contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

DOIUSEND

EZA6050E **Error *rc* receiving on DLC path**

Explanation

An error was encountered while receiving data using a DLC path from the TCPIP address space. The DLC path is severed. The indicated return code is the code returned using register 15 from the receive attempt.

System action

Processing continues.

Operator response

Check that the TCPIP address space is active.

System programmer response

Check the TCPIP log data set for an indication of the nature of the error. Attempt to reestablish the IUCV path by restarting the SNALINK LU6.2 device from within TCPIP. If the error persists, contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

DOIURECV

EZA6051E **Error *rc* severing DLC path**

Explanation

An error was encountered while severing a DLC path between the TCPIP address space and the SNALINK LU6.2 address space. The indicated return code is the code returned using register 15 from the sever attempt.

System action

Processing continues as though the DLC path was successfully severed.

Operator response

Check that the TCPIP address space is active.

System programmer response

Check the TCPIP log data set for an indication of the nature of the error. If problems are encountered in the subsequent reestablishment of the path, shut down, and restart the SNALINK LU6.2 address space. If the error persists, contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

DOIUSEVR

EZA6052E **Error *rc* in DLC initialization**

Explanation

An error was encountered during the DLC initialization. The indicated return code is the code returned using register 15 from the initialization routine.

System action

The SNALINK LU6.2 address space shuts down immediately.

Operator response

Check that the TCPIP address space is active.

System programmer response

Check the TCPIP log data set for an indication of the nature of the error. Restart the SNALINK LU6.2 address space. If the error persists, contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

IUCVINIT

EZA6059W **Error encountered in processing *name* - dataset closed**

Explanation

An I/O error was detected during the processing of the data set defined by the indicated ddname. This message is accompanied by message EZA6060W, which describes the nature of the error detected.

System action

Message EZA6060W is output to indicate the nature of the error detected, and the data set is closed.

Operator response

No action is needed.

System programmer response

Locate the accompanying message EZA6060W to identify the nature of the error.

Module

LU62IO

Procedure name

OPENOUT, OPENINP, PUTDS, GETDS, CLOSEDS

EZA6060W ***I/O_error_message***

Explanation

This message accompanies message EZA6059W; the indicated error message describes the nature of the I/O error detected.

System action

See message EZA6059W.

Operator response

No action is needed.

System programmer response

The indicated error message was constructed using the SYNADAF system macro. See the documentation for this macro to determine the exact nature of the I/O error encountered. Correct the error, and restart the SNALINK LU6.2 address space.

Module

LU62IO

Procedure name

OPENOUT, OPENINP, PUTDS, GETDS, CLOSEDS

EZA6061W **Abend *code-rc* trapped in *operation* for *name* - dataset closed**

Explanation

An abend condition was encountered during the running of the indicated operation on the data set defined by the indicated ddname. The abend was trapped and the address space allowed to continue.

System action

Processing continues; the data set is closed by the system.

Operator response

No action is needed.

System programmer response

Determine the nature of the abend condition using the indicated Abend code and reason code values, correct the problem, and restart the SNALINK LU6.2 address space.

Module

LU62IO

Procedure name

OPENOUT, OPENINP, PUTDS, GETDS, CLOSEDS

EZA6062W **Error in open of *name* - output to dataset will be suppressed**

Explanation

An error was encountered during the opening of the output data set defined by the indicated ddname. All attempts to write to the data set are suppressed.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

See previous error messages to determine the nature of the problem, correct the problem, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

L62ENTRY

EZA6063W Error in open of *name* - no data will be read

Explanation

An error was encountered during the opening of the input data set defined by the indicated ddname.

System action

Processing continues as though the input data set was empty.

Operator response

No action is needed.

System programmer response

See previous error messages to determine the nature of the problem, correct the problem, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

L62ENTRY

EZA6064W Record Format of *name* must be either F or V - dataset closed

Explanation

An unexpected record format was encountered during the open of the input data set defined by the indicated ddname. Only fixed- or variable-length record formats are supported.

System action

Processing continues as though the input data set was empty.

Operator response

No action is needed.

System programmer response

Correct the record format of the data set, and restart the SNALINK LU6.2 address space.

Module

LU62IO

Procedure name

GETDS

EZA6065W **Error in read of *name* - dataset closed**

Explanation

An error was encountered during a read operation on the data set defined by the indicated ddname.

System action

Processing continues as though an end-of-file condition was encountered.

Operator response

No action is needed.

System programmer response

See previous error messages to determine the nature of the problem, correct the problem, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

READTCPD, RLCFG#P1, RLCFG#P2

EZA6074E **MODIFY sub-command missing**

Explanation

A MODIFY command was entered for the SNALINK LU6.2 address space from an operator console, but no subcommand was specified. The command is ignored.

System action

Processing continues.

Operator response

Enter the MODIFY command in the correct format.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

MODIFY

EZA6075E **Invalid MODIFY sub-command**

Explanation

A MODIFY command was entered for the SNALINK LU6.2 address space from an operator console, but the subcommand specified was not recognized by the SNALINK LU6.2 address space. The command is ignored.

System action

Processing continues.

Operator response

Enter the MODIFY command in the correct format.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

MODIFY

EZA6076W **DROP: No Current Connection to IP Address *address***

Explanation

A DROP subcommand, which included the indicated IP address in the specified address range, was entered from an operator console (using the MODIFY console command), but there was no current connection to that destination. The rest of the connections in the address range are closed.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROPCONN

EZA6077W **RESTART: Connection to IP Address *address* Already Established**

Explanation

A RESTART subcommand, which included the indicated IP address in the specified address range, was entered from an operator console (using the MODIFY console command), but there was already a connection established with that destination. Connections are established with the rest of the destinations in the address range.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

STRTCONN

EZA6078E **DROP: IP/LU/ALL Expected after DROP**

Explanation

The syntax of a DROP subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP

EZA6079E ***command*: '=' Expected after IP**

Explanation

The syntax of the indicated subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6080E ***command*: IP Address Expected after 'IP='**

Explanation

The syntax of the indicated subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6081E ***command*: Invalid Format for a fully qualified IP Address**

Explanation

The IP address contained in the indicated subcommand entered from an operator console (using the MODIFY console command) was not in the required format. The IP address should be in the correct dotted-decimal format for a network node (for example, 192.9.207.100). The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6082E ***command*: IP Address Not Defined in Configuration Dataset**

Explanation

The IP address contained in the indicated subcommand entered from an operator console (using the MODIFY console command) is not defined as a destination address in the SNALINK LU6.2 configuration data set. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using a valid IP address.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6083E ***command*: '=' Expected after LU**

Explanation

The syntax of the indicated subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6084E ***command*: LU Name Expected after 'LU='**

Explanation

The syntax of the indicated subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6085E ***command*: Invalid Format for LU Name**

Explanation

The LU name contained in the indicated subcommand entered from an operator console (using the MODIFY console command) was longer than 8 characters. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6086E ***command*: LU Name Not Defined in Configuration Dataset**

Explanation

The LU name contained in the indicated subcommand entered from an operator console (using the MODIFY console command) is not defined in the SNALINK LU6.2 configuration data set. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using a valid LU name.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

DROP, LIST, RESTART

EZA6087E **RESTART: Invalid Range Specified - IP,LU,INIT or ALL expected**

Explanation

The syntax of a RESTART subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

RESTART

EZA6088E **LIST: Invalid Range Specified - IP,LU,ACTIVE or ALL expected**

Explanation

The syntax of a LIST subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

LIST

EZA6089E **TRACE: ON/OFF/DETAIL or ALL/IP expected after TRACE**

Explanation

The syntax of a TRACE subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA6090E **TRACE: ALL/IP expected after ON/OFF/DETAIL**

Explanation

The syntax of a TRACE subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA6091E **TRACE: '=' expected after IP**

Explanation

The syntax of a TRACE subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA6092E **TRACE: IP address expected after '='**

Explanation

The syntax of a TRACE subcommand entered from an operator console (using the MODIFY console command) was incorrect. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA6093E TRACE: Invalid format for fully qualified IP address

Explanation

The IP address contained in a TRACE subcommand entered from an operator console (using the MODIFY console command) was not in the required format. The IP address should be in the correct dotted-decimal format for a network node (for example, 192.9.207.100). The command is ignored.

System action

Processing continues.

Operator response

Enter the command using the correct syntax.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA6094E TRACE: IP address not defined in Configuration dataset

Explanation

The IP address contained in a TRACE subcommand entered from an operator console (using the MODIFY console command) is not defined as a destination IP address in the SNALINK LU6.2 configuration data set. The command is ignored.

System action

Processing continues.

Operator response

Enter the command using a valid IP address.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

TRACE

EZA6095W **RESTART: No connections to start**

Explanation

A RESTART subcommand, with a range specification of INIT, was entered from an operator console (using the MODIFY console command), but there are no destinations defined with the INIT option in the configuration data set. The command is ignored.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

LU62MDFY

Procedure name

RESTART

EZA6114E **LU62CFG: Line *number* Col *column*: Invalid Statement Type**

Explanation

An incorrect statement type was encountered in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Remove or correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6115E **LU62CFG: Line *number* Col *column*: VTAM statement - Application Id expected**

Explanation

The application ID was omitted from a VTAM statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6116E **LU62CFG: Line *number* Col *column*: VTAM statement - Invalid format for Application Id**

Explanation

The application ID contained in a VTAM statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is longer than 8 characters. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6117E **LU62CFG: Line *number* Col *column*: VTAM statement - Invalid format for Password**

Explanation

The password contained in a VTAM statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is longer than 8 characters. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6118E **LU62CFG: Line *number* Col *column*: LINK statement - TCP/IP Link Name expected**

Explanation

The TCPIP link name field of a LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is missing. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6119E **LU62CFG: Line *number* Col *column*: LINK statement - Invalid TCP/IP Link Name**

Explanation

The TCPIP link name contained in a LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is longer than 8 characters. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6120E **LU62CFG: Line *number* Col *column*: LINK statement - Link Name already defined**

Explanation

The TCPIP link name field in a LINK statement in the SNALINK LU6.2 configuration data set has already been defined in a previous LINK statement. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6121E **LU62CFG: Line *number* Col *column*: LINK statement - Log Mode Table Name expected**

Explanation

The Log Mode Table field of a LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is missing. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6122E **LU62CFG: Line *number* Col *column*: LINK statement - Invalid Log Mode Table Name**

Explanation

The Log Mode Table name contained in a LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is longer than 8 characters. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6123E **LU62CFG: Line *number* Col *column*: LINK statement - Invalid Idle Timeout value**

Explanation

The Idle Timeout value contained in a LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is not a valid integer in the range 0 to 2^{31}-1. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6124E **LU62CFG: Line *number* Col *column*: DEST statement - Destination IP Address expected**

Explanation

The Destination IP address field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is missing. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6125E **LU62CFG: Line *number* Col *column*: DEST statement - Invalid format for a fully qualified IP address**

Explanation

The Destination IP address field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is not in the required format. The IP address should be in the correct dotted-decimal format for a network node (for example, 192.9.207.100). The error was detected at the indicated line of the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6126E **LU62CFG: Line *number* Col *column*: DEST statement - Duplicate IP address detected**

Explanation

The Destination IP address field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG has already been defined in a previous DEST statement. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct or remove the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6127E **LU62CFG: Line *number* Col *column*: DEST statement - Send LU Name expected**

Explanation

The Send LU Name field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is missing. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6128E **LU62CFG: Line *number* Col *column*: DEST statement - Invalid Send LU Name**

Explanation

The Send LU Name field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is longer than 8 characters. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6129E	**LU62CFG: Line *number* Col *column*: DEST statement - Receive LU Name or '=' expected**

Explanation

The Receive LU Name field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is missing. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6130E	**LU62CFG: Line *number* Col *column*: DEST statement - Invalid Receive LU Name**

Explanation

The Receive LU Name field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is longer than 8 characters. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6131E **LU62CFG: Line *number* Col *column*: DEST statement - Duplicate LU Name Detected**

Explanation

The Send or Receive LU Name field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG has already been defined on a previous DEST statement. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6132E **LU62CFG: Line *number* Col *column*: DEST statement - LU Name cannot equal the Application Id on the VTAM statement**

Explanation

The Send or Receive LU Name field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is the same as that defined on the VTAM statement. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6133E **LU62CFG: Line *number* Col *column*: DEST statement - Invalid Connection Start Type**

Explanation

The Connection Start Type field of a DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG contains an incorrect value. The Connection Start Type value should be either DATA, INIT, or blank. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6134E **LU62CFG: Line *number* Col *column*: BUFFERS statement - Maximum Packet Size value expected**

Explanation

The Maximum Packet Size field of a BUFFERS statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is missing. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6135E **LU62CFG: Line *number* Col *column*: BUFFERS statement - Invalid Maximum Packet Size value**

Explanation

The Maximum Packet Size value contained in a BUFFERS statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is not a valid integer in the range 20 - 32758. The error was detected at the indicated line of the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6136E **LU62CFG: Line *number* Col *column*: BUFFERS statement - Invalid number of Additional Send Buffers**

Explanation

The Additional Send Buffers value contained in a BUFFERS statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is not a valid integer in the range 0 to 2^{31}-1. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6137E **LU62CFG: Line *number* Col *column*: BUFFERS statement - Invalid Send Queue Limit value**

Explanation

The Send Queue Limit value contained in a BUFFERS statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is not a valid integer in the range 0 to 2^{31}-1. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6138E **LU62CFG: Line *number*: TRACE statement - ON/OFF/DETAIL or ALL/IP expected after TRACE**

Explanation

The syntax of a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is incorrect. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6139E **LU62CFG: Line *number*: TRACE statement - ALL/IP expected after ON/OFF**

Explanation

The syntax of a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is incorrect. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6140E **LU62CFG: Line *number*: TRACE statement - '=' expected after IP**

Explanation

The syntax of a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is incorrect. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6141E **LU62CFG: Line *number*: TRACE statement - IP address expected after '='**

Explanation

The syntax of a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is incorrect. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6142E **LU62CFG: Line *number*: TRACE statement - Invalid format for a fully qualified IP address**

Explanation

The IP address field of a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG is not in the required format. The IP address should be in the correct dotted-decimal format for a network node (for example, 192.9.207.100). The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6143E **LU62CFG: Line *number*: TRACE statement - IP address not defined**

Explanation

The IP address specified in a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG was not previously defined in a DEST statement. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6144E **LU62CFG: Line *number*: TRACE statement - Extraneous trailing data on line**

Explanation

Extra trailing data was found on a line containing a TRACE statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P2

EZA6145E **LU62CFG: Line *number*: Only one VTAM statement may be specified**

Explanation

More than one VTAM statement was found in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Remove one of the VTAM statements, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6146E **LU62CFG: VTAM statement missing**

Explanation

A VTAM statement was not found in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Add a VTAM statement to the configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6147E **LU62CFG: Line *number*: Only one BUFFERS statement may be specified**

Explanation

More than one BUFFERS statement was found in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Remove one of the BUFFERS statements, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6148E **LU62CFG: BUFFERS statement missing**

Explanation

A BUFFERS statement was not found in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Add a BUFFERS statement to the configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6149E **LU62CFG: No LINK statements defined**

Explanation

No LINK statements were found in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Add a LINK statement for each IUCV connection required between the SNALINK LU6.2 address space and the TCPIP address space, and restart the SNALINK LU6.2 address space.

Each of these connections should also be defined in the *hlq*.PROFILE.TCPIP data set using DEVICE and LINK statements. If DEVICE and LINK statements are added to the *hlq*.PROFILE.TCPIP data set, restart the TCPIP address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6150E **LU62CFG: No DEST statements defined**

Explanation

No DEST statements were found in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Add a DEST statement for each destination IP address in the network directly connected to the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6151E **LU62CFG: Line *number*: No DEST statements defined for previous LINK statement**

Explanation

No DEST statements were found for the previous LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Either add a DEST statement following the previous LINK statement, or delete the previous LINK statement; then restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6152E **LU62CFG: Line *number*: VTAM statement must appear before first LINK statement**

Explanation

No VTAM statement was found before the first LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6153E **LU62CFG: Line *number*: A DEST statement can only appear after a LINK statement**

Explanation

No LINK statement was found before the first DEST statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the configuration data set, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6154E **LU62CFG: No DEST statements defined for last LINK statement**

Explanation

No DEST statements were found for the last LINK statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Either add a DEST statement following the last LINK statement or delete the last LINK statement; then restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1

EZA6155E **LU62CFG: Line *number* Col *column*: Extraneous trailing data on line**

Explanation

Extra trailing data was found on a line following a statement in the SNALINK LU6.2 configuration data set defined by the ddname LU62CFG. The error was detected at the indicated line in the indicated column.

System action

The SNALINK LU6.2 address space shuts down immediately after the current pass of the data set is completed.

Operator response

No action is needed.

System programmer response

Correct the line in error, and restart the SNALINK LU6.2 address space.

Module

LU62INIT

Procedure name

RLCFG#P1, RLCFG#P2

EZA6170W **Unable to setup connection entry.**

Explanation

An error occurred during the creation of a Connection Table entry for a new connection. The cause is either insufficient storage available to create the buffers associated with the entry, or a program logic error. Preceding messages in the SYSPRINT data set indicate the exact nature of the error.

System action

Processing continues; the connection attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

If preceding messages indicate insufficient storage, increase the REGION size parameter for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62VTAM

Procedure name

ALOCCONN

EZA6171W **Unable to start connection for *address*; connection entry setup failed**

Explanation

This message is output on the console when a connection establishment that was initiated using the MODIFY command has been unsuccessful because of an error in creating the corresponding Connection Table entry. This error can occur when either there is insufficient storage available to create the buffers associated with the entry or a program logic error has occurred; preceding messages in the SYSPRINT data set indicate the exact nature of the error.

System action

Processing continues; the connection attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

If preceding messages indicate insufficient storage, increase the REGION size parameter for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62VTAM

Procedure name

MAKECONN

EZA6172E **Sequential search length too large *length***

Explanation

The indicated search field length passed to the sequential search routine was longer than 255.

System action

Processing continues as though the required entry was not found. Depending on the type of entry being located, this can lead to subsequent errors.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62MAIN

Procedure name

SEQSERCH

EZA6173E **Binary search length too large *length***

Explanation

The indicated search field length passed to the binary search routine was longer than 255.

System action

Processing continues as though the required entry was not found. Depending on the type of entry being located, this can lead to subsequent errors.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62MAIN

Procedure name

BINSERCH

EZA6174W **Unable to setup DLC table entry for *link*; entry already active for path**

Explanation

An attempt to establish a DLC connection for the indicated TCPIP link has been unsuccessful because the corresponding DLC table entry is already marked as active.

System action

Processing continues; the DLC connection attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center. The data structures can be reinitialized by restarting the SNALINK LU6.2 address space.

Module

LU62IUCV

Procedure name

SETUPIUC

EZA6175E **Unable to setup DLC table entry; free memory exhausted, RC=*rc***

Explanation

An attempt to establish a DLC connection has been unsuccessful because of an error in a GETMAIN macro invocation.

System action

Processing continues; the DLC connection attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

Check the indicated GETMAIN return code value to determine the cause of the error. If insufficient storage caused the error, increase the REGION size parameter for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62IUCV

Procedure name

SETUPIUC

EZA6176W **Unable to clear DLC table entry; already inactive for path**

Explanation

An attempt to close a DLC connection for the indicated TCPIP link has been unsuccessful because the corresponding DLC table entry is already marked as inactive.

System action

Processing continues; the DLC connection is assumed to be correctly closed.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

CLEARIUC

EZA6177E **Unable to clear DLC table entry for path; can’t free memory RC=*rc***

Explanation

An attempt to close a DLC connection has been unsuccessful because of an error in a FREEMAIN macro invocation. The indicated return code is the return code for the unsuccessful FREEMAIN macro invocation.

System action

Processing continues; the DLC connection is assumed to be correctly closed.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

CLEARIUC

EZA6178W **The connection entry passed to SETUPCON is already in use**

Explanation

An attempt to establish a connection with a destination node has been unsuccessful because of the corresponding Connection Table entry already being marked as active.

System action

Processing continues; the connection establishment attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

SETUPCON

EZA6179E **Getmain error RC=** ***rc***

Explanation

An attempt to establish a connection with a destination node has been unsuccessful because of an error in a GETMAIN macro invocation. The indicated return code is the return code value from the unsuccessful GETMAIN macro invocation.

System action

Processing continues; the connection establishment attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

Check the indicated GETMAIN return code value to determine the cause of the error. If insufficient storage caused the error, increase the REGION size parameter for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62VTAM

Procedure name

SETUPCON

EZA6180E **GENCB error RC= *rc***

Explanation

An attempt to establish a connection with a destination node has been unsuccessful because the VTAM RPL could not be generated in the Connection Table entry. The indicated return code is the return code value from the unsuccessful GENCB macro invocation.

System action

Processing continues; the connection establishment attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

Check the indicated GENCB return code value to determine the cause of the error, and correct the problem if possible.

Module

LU62VTAM

Procedure name

SETUPCON

EZA6181W **The connection entry passed to CLEARCON is already inactive**

Explanation

The Connection Table entry address, passed to the routine responsible for clearing the entry and freeing the memory used by associated buffers, is already marked as inactive.

System action

Processing continues; the connection is assumed to be correctly closed.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

CLEARCON

EZA6182E **Freemain error; RC= *rc***

Explanation

An attempt to close a connection with a destination node has been unsuccessful because of an error in a FREEMAIN macro invocation. The indicated return code is the return code for the unsuccessful FREEMAIN macro invocation.

System action

Processing continues; the connection is assumed to be correctly closed.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

CLEARCON

EZA6183E **The IP address *address* from the connection table is undefined**

Explanation

An attempt to close a connection with a destination node has been unsuccessful because of a probable program logic error or internal data structure corruption.

System action

Processing continues; the connection is assumed to be correctly closed.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

KILLCONN

EZA6184E **Invalid connection entry address passed to ADD INACTIVITY**

Explanation

An error occurred while adding a connection to the timeout inactivity list; the connection entry address did not reference a valid Connection Table entry.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62MAIN

Procedure name

ADDINACT

EZA6185E **Invalid connection entry address passed to SUB INACTIVITY**

Explanation

An error occurred while removing a connection from the timeout inactivity list; the connection entry address did not reference a valid Connection Table entry.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62MAIN

Procedure name

SUBINACT

EZA6186E **The inactivity queue has lost the *address* connection**

Explanation

The Connection Table entry for the indicated IP address has been lost from the timeout inactivity list.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62MAIN

Procedure name

SUBINACT

EZA6187E **Unable to match RPL address to send or receive for *address***

Explanation

The RPL address returned from VTAM for a conversation allocation request did not match either the send or receive RPL address.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Probable program logic error or data corruption in either VTAM or SNALINK LU6.2. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMALOC

EZA6188W **The connection entry passed to KILLCONN is already inactive**

Explanation

An attempt to close a connection with a destination node has been unsuccessful because of the corresponding Connection Table entry already being marked as inactive.

System action

Processing continues; the connection is assumed to be correctly closed.

Operator response

No action is needed.

System programmer response

Probable program logic error. Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

KILLCONN

EZA6189W **Unable to honor connection request for *address*; connection entry setup failed**

Explanation

An error was encountered during the creation of a Connection Table entry for a new connection that was initiated from the destination node associated with the indicated IP address. This error can occur when either there is not enough storage to create the buffers associated with the entry or a program logic error has occurred. The preceding messages in the SYSPRINT data set indicate the exact nature of the error.

System action

Processing continues; the connection attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

If preceding messages indicate insufficient storage, increase the REGION size parameter for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6190W **Program Parameters Ignored**

Explanation

Program parameters were passed to the SNALINK LU6.2 address space, but no parameters are processed by the SNALINK LU6.2 address space. The parameters are ignored.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Remove the program parameters from the invocation JCL.

Module

LU62INIT

Procedure name

L62ENTRY

EZA6192W **CIB for START Command not freed - MODIFY Commands may not be available**

Explanation

An error occurred during the initialization of MODIFY command handling. MODIFY commands might not be available.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Probable program error. Contact the IBM Software Support Center.

Module

LU62MAIN

Procedure name

MAINPROC

EZA6193W **Unrecognized Console Command - Command Ignored**

Explanation

An unrecognized console command was passed to the SNALINK LU6.2 address space; only START, MODIFY, STOP, and PURGE commands are recognized. The command is ignored.

System action

Processing continues.

Operator response

Enter the required command using the correct format.

System programmer response

No action is needed.

Module

LU62MAIN

Procedure name

MAINPROC

EZA6204W **Datagram for *address* discarded, queue limit exceeded**

Explanation

The length of the send queue, which holds IP datagrams waiting to be passed to VTAM, has exceeded the maximum length specified in the BUFFERS statement in the SNALINK LU6.2 configuration data set. The indicated address is the destination IP address of the datagram triggering this message. The datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Consider increasing the send queue limit or maximum packet size value on the BUFFERS statement. If possible, tune the VTAM connection with the destination node to provide better throughput.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA6205W **Datagram for *address* discarded, no free memory for connection entry**

Explanation

An IP datagram for the indicated IP address has initiated a connection attempt with the corresponding destination node, but the connection attempt was unsuccessful because of insufficient available storage to allocate the required buffers. The datagram is discarded.

System action

Processing continues; the connection attempt is abnormally ended.

Operator response

No action is needed.

System programmer response

Increase the value of the REGION parameter for the SNALINK LU6.2 address space, and restart the address space.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA6206W **Datagram for *address* discarded, VTAM send buffer pool exhausted**

Explanation

The buffer pool used to hold IP datagrams waiting to be passed to VTAM has been exhausted. The datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Consider increasing the number of additional send buffers and maximum packet size values in the BUFFERS statement. If possible, tune the VTAM connections to provide better throughput.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA6207W **VTAM send queue limit exceeded for *address***

Explanation

The length of the send queue, which holds IP datagrams waiting to be passed to VTAM, has exceeded the maximum length specified in the BUFFERS statement in the SNALINK LU6.2 configuration data set. The indicated address is the destination IP address of the datagram triggering this message. Message EZA6204W is also written to the SYSPRINT data set, and the datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

See message EZA6204W.

Module

LU62VTAM

Procedure name

ADDVTSND

EZA6208W **Datagram for *address* discarded, *link* requested, *link* defined**

Explanation

An IP datagram for the indicated IP address was received from TCPIP using the indicated link. However, in the SNALINK LU6.2 configuration data set, the first indicated link has been defined to use the second indicated link. The datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Correct the *hlq*.PROFILE.TCPIP data set or the SNALINK LU6.2 configuration data set, and restart the address spaces.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA6209W **Datagram for *address* discarded, link *link* closed**

Explanation

An IP datagram has been received from the indicated IP address, but the associated IUCV connection on which the datagram is to be passed to TCPIP is not currently open. TCPIP is responsible for initiating the establishment of IUCV connections and will try to establish the connections again every 30 seconds. The datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the TCPIP address space is active and that the IUCV connection to the SNALINK LU6.2 address space has been successfully started.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA6210W **Datagram for *address* discarded, IP address not defined**

Explanation

An IP datagram with the indicated first-hop address was received from TCPIP (using IUCV), but the address is not defined in the SNALINK LU6.2 configuration data set. The datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Correct the SNALINK LU6.2 configuration data set, and restart the address space.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA6211W **Message from *link* discarded, receive buffer too small**

Explanation

A message sent by the TCPIP address space using IUCV is too large to be received into the IUCV receive buffer. The indicated link name is the TCPIP link name on which the message was sent. The datagrams in the message are discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Probable program error. Contact the IBM Software Support Center.

Module

LU62IUCV

Procedure name

DOIURECV

EZA6212W **Datagram for *address* discarded, Send datagram exceeded the MAX packet size**

Explanation

An IP datagram received from TCPIP (using IUCV) is larger than the maximum packet size value defined in the BUFFERS statement in the SNALINK LU6.2 configuration data set. This maximum packet size value should be set to the same value as that defined for the corresponding link in the *hlq*.PROFILE.TCPIP data set. The datagram is discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Correct the *hlq*.PROFILE.TCPIP data set or the SNALINK LU6.2 configuration data set, and restart the address spaces.

Module

LU62IUCV

Procedure name

IUCVPMES

EZA6213W **Datagram from *address* discarded, Receive datagram exceeded the MAX packet size**

Explanation

An IP datagram sent from a destination node (using VTAM) is larger than the maximum packet size value defined in the BUFFERS statement in the SNALINK LU6.2 configuration data set. This maximum packet size value should be set to the same value on all nodes in the network directly connected to the SNALINK LU6.2 address space. The datagram is received in sections and then discarded.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

Check that the maximum packet size value is set to the same value on all nodes in the network directly connected to the SNALINK LU6.2 address space.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA6219E **IUCV Initialization Failed, VMCF Subsystem not located**

Explanation

While trying to determine the Program Call number for use by the IUCV interface, the JES Subsystem Communication Vector Table (SSCT) entry for VMCF could not be located.

System action

The SNALINK LU6.2 address space shuts down immediately.

Operator response

No action is needed.

System programmer response

Verify that the VMCF subsystem is active. See Step 3: Configure VMCF and TNF in z/OS Communications Server: IP Configuration Guide for more information about starting VMCF and TNF. Check for previous operator messages that might indicate why VMCF failed to initialize.

Module

LU62INIT

Procedure name

GETIPC

EZA6220S **Unable to start connection for *address*; no free connection entries**

Explanation

An attempt to establish a connection with the destination node associated with the indicated IP address failed because there were no free Connection Table entries. This indicates a program logic error or internal data corruption has occurred because there should have been sufficient entries allocated during initialization of the address space.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

ALOCCONN

EZA6221S **Invalid exit type found on VTAM event queue, Type=** ***type***

Explanation

The indicated exit type was found on the VTAM event queue. The valid exit types are SEND, RECV, ATTN, ALOC, and DALC.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

PROCVTAM

EZA6222S **No connection entry found in RPL record**

Explanation

All VTAM requests are made with the address of the relevant Connection Table entry in the word immediately preceding the RPL address defined to VTAM. This address has somehow been corrupted.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMALOC, VTAMSEND, VTAMRECV, VTAMDALC

EZA6223S **Unmatched RPL address for *address* in DEALOC**

Explanation

The RPL address returned from a VTAM deallocate request does not match either the Send or Receive RPL address, which indicates that the RPL has probably been corrupted.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMDALC

EZA6224S **VTAM send completed, but send queue empty for *address***

Explanation

The VTAM send queue was empty when it was expected to contain an entry, indicating that the queue has probably been corrupted.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMSEND

EZA6225S **ATTN Exit event type invalid. Type= *type***

Explanation

The indicated ATTN exit type was detected. The valid exit types are FMH5, CNOS, and LOSS.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMATTN

EZA6226S **Negative send queue count for *address***

Explanation

A counter, which contains the number of entries in the VTAM send, contains a negative number, indicating a probable program logic error.

System action

The SNALINK LU6.2 address space issues a user abend immediately.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

SUBVTSND

EZA6227S **Bad RPL specified on *command***

Explanation

The return codes from the indicated VTAM APPC command show that the RPL specified was not valid. Messages EZA6029E to EZA6032E are written to the SYSPRINT data set to provide further details of the error.

System action

The SNALINK LU6.2 address space issues a user abend.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

ANALYZE

EZA6228S **Bad RPLX specified on *command***

Explanation

The return codes from the indicated VTAM APPC command show that the extended RPL specified was not valid. Messages EZA6029E to EZA6032E are written to the SYSPRINT data set to provide further details of the error.

System action

The SNALINK LU6.2 address space issues a user abend.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

ANALYZE

EZA6229S ***command* received a policing error**

Explanation

The return codes from the indicated VTAM APPC command indicate that a non-APPC command was issued on an APPC connection. Messages EZA6029E to EZA6032E are written to the SYSPRINT data set to provide further details of the error.

System action

The SNALINK LU6.2 address space issues a user abend.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

ANALYZE

EZA6230S **Pending message buffer pool exhausted**

Explanation

The free pool for the pending message queue has been exhausted. This should never happen because sufficient entries are allocated during initialization, and a maximum of one entry can be used by each connection. This is a probable program logic error. A dump is produced if the ddname SYSUDUMP has been defined.

System action

The SNALINK LU6.2 address space abnormally ends.

Operator response

No action is needed.

System programmer response

Contact the IBM Software Support Center.

Module

LU62VTAM

Procedure name

VTAMRECV

EZA6275I **SNMP query engine running and awaiting queries...**

Explanation

The SNMP query engine is running and it is available to handle queries.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

sqeserv

Procedure name

MAIN

EZA6276I **There are *available_FD's* Client connections possible**

Explanation

The number of client connections possible is determined by subtracting the number of current connections from the maximum number of connections defined in the management information base (MIB).

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

sqeserv

Procedure name

MAIN

EZA6277S **more than *trans_count* transactions in a second**

Explanation

The maximum number of transactions per second has been exceeded. This can be a sign that SQESERV is in a loop.

System action

The SNMP query engine exits.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

sqeserv

Procedure name

await_queries

EZA6278I **await_queries : select**

Explanation

The SQE server has attempted to issue a select function to determine if any queries were outstanding. Message EZA6279E is displayed.

System action

The SQE server ends.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

await_queries

EZA6279E **select failed in await_queries**

Explanation

The SQE server encountered an error issuing a select function call to determine if any queries were outstanding.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

await_queries

EZA6280E **accept_new_client : accept**

Explanation

The SQE server encountered an error while trying to establish an accept for a TCP listen.

System action

SNMP query server continues processing.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

accept_new_client

EZA6281E **mkNVlisten: socket**

Explanation

The SQE server encountered an error while trying to establish a socket for a TCP listen.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

mkNVlisten

EZA6282E **mkNVlisten: bind**

Explanation

The SQE server encountered an error while trying to establish a bind for a TCP listen.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

mkNVlisten

EZA6283E **mkNVlisten: socket(AF_IUCV)**

Explanation

The SQE server encountered an error while trying to establish an IUCV socket for a TCP listen.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

mkNVlisten

EZA6285E **Can’t make UDP port - can't send SNMP requests**

Explanation

The SQE server could not make the SNMP UDP port.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

This error message is preceded by a message indicating an error with either the socket or bind request.

Module

SQESERV

Procedure name

create_ports

EZA6286W **Can’t make SNMP trap port - can't receive traps**

Explanation

The SQE server cannot make the SNMP trap port because of an unsuccessful socket call or bind.

System action

The SQESERV module ends.

Operator response

Check that another SQESERV program is not already running. Only one SQESERV program can use the SNMP trap port. If this is the only SQESERV program attempting to run, contact the system programmer.

System programmer response

Check that the user is allowed to use the reserved port by making the appropriate PORT entries in the *hlq*.PROFILE.TCPIP data set.

Module

SQESERV

Procedure name

create_ports

EZA6287E **Can’t do TCP listen - can't accept TCP clients**

Explanation

The SQE server encountered an error attempting a TCP listen.

System action

The SQESERV module ends with exit code 1.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

create_ports

EZA6288E **Can’t do IUCV listen - can't accept IUCV clients**

Explanation

The SQE server encountered an error attempting an IUCV listen.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

create_ports

EZA6289W **Can’t make raw ICMP socket - can't do PING**

Explanation

The SQE server encountered an error making the PING port. SNMP Ping requests are unsuccessful.

System action

Processing continues without the raw socket.

Operator response

Check that the user is permitted to use RAW sockets.

System programmer response

If you are running on a system with the RACF program installed, check that the SNMPQE user ID is authorized to the RACF. See the EZARACF sample shipped in SEZAINST for sample RACF definitions.

Restart SNMPQE with debug level 4 to get detailed information about the problem. See the information about specifying SNMPQE parameters in z/OS Communications Server: IP Configuration Reference for more information about starting SNMPQE with debug level 4.

Module

SQESERV

Procedure name

create_ports

EZA6290I **Accepted new client connection**

Explanation

The SQE server was able to establish a new client connection.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

accept_new_client

EZA6292I **Received following NVquery packet.**

Explanation

The SQE server was able to receive a packet. Following this message will be a hexadecimal display of the packet.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

getNVquery

EZA6293I **Terminated client connection.**

Explanation

The SQE server has ended a client connection.

System action

The client connection is ended and the SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

terminate_connection

EZA6294E **getSNMPresponse: recvfrom**

Explanation

The SQE server was unable to receive a response from an SNMP query.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Verify that the SNMPD agent is available.

Module

SQESERV

Procedure name

getSNMPresponse

EZA6295I **Received following SNMP_response packet**

Explanation

The SQE server was able to receive a response from an SNMP query. The hexadecimal display of the packet will follow.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

getSNMPresponse

EZA6297E **error code *err_code* : out of memory**

Explanation

The SQE server was unable get enough storage.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Increase the region size and restart the SNMP query engine.

Module

SQESERV, SQEQUEUE, SQEPNV, SQEMKREQ

Procedure name

Various

EZA6298E **create_SNMP_TRAP_port : socket**

Explanation

The SQE server was unable to establish a socket for an SNMP trap port.

System action

The SQESERV module ended.

Operator response

Tell the system programmer about the error.

System programmer response

Verify that the port for SNMPQE was defined correctly in the *hlq*.PROFILE.DATA data set. If the port was defined correctly see z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

create_SNMP_trap_port

EZA6299E **create_SNMP_TRAP_port : bind**

Explanation

The SQE server was unable to bind a socket for an SNMP trap port.

System action

The SQESERV module ended.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

create_SNMP_trap_port

EZA6300I **getSNMPtrap : recvform**

Explanation

The SQE server was unable to receive an SNMP trap packet.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

getSNMPtrap

EZA6301I **Received following SNMP_trap packet**

Explanation

The SQE server was able to receive a trap packet for the SNMP query server. The hexadecimal display of the packet will follow.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

getSNMPtrap

EZA6303E **recvPING: recvfrom**

Explanation

The SQE server was unable to receive a SNMP ICMP packet.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for information about socket errors.

Module

SQESERV

Procedure name

create_SNMP_trap_port

EZA6304E **Error code *err_code* : received ICMP packet >= *size* bytes**

Explanation

The SQE server was unable to receive an SNMP ICMP packet because the packet was too large. The largest packet size supported by the SQE server is 4096 bytes.

System action

Processing continues.

Operator response

Reissue the PING request with a smaller packet size.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

getSNMPresponse

EZA6305I **dumping packet of *size* bytes**

Explanation

The SQE server will print a packet as hexadecimal data.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQESERV

Procedure name

hex_dump

EZA6306W **resolver not yet supported**

Explanation

The SQE server found a Q_RESOLVER_RESPONSE request, which is not currently supported. The request is ignored.

System action

Processing continues with the next request.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEQUEUE

Procedure name

queue_entry

EZA6307E **queue_entry**

Explanation

Either an SNMP, RESOLVER, PING, or an incorrect query was requested and was unsuccessful.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Determine the cause of the error by referring to the messages that follow.

Module

SQEQUEUE

Procedure name

queue_entry

EZA6308I **sending SNMP request to *host***

Explanation

The SQE query engine is sending a request to the host specified.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEQUEUE

Procedure name

send_SNMP_request

EZA6309W **ret_response:cant find q_ptr .**

EZA6310W **queue= *queue* req_id= *req_id* minor_err= *minor_err***

Explanation

The SQE query engine has no more outstanding responses in the queue.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEQUEUE

Procedure name

ret_response

EZA6311W **fd *fd* not ready for writing for *timeout* seconds**

Explanation

The SQE query engine timed out while trying to write a response to the connection.

System action

The connection is closed.

Operator response

Reestablish the connection.

System programmer response

Verify that the SNMPD client is available.

Module

SQEQUEUE

Procedure name

write_reply

EZA6312W **Dropping packet for client on fd= *fd***

Explanation

The SQE query engine could not write a response to the connection.

System action

Connection is closed.

Operator response

Contact the system programmer.

System programmer response

Determine if the agent is available, then reestablish the connection.

Module

SQEQUEUE

Procedure name

ret_response

EZA6313W **Ignored write_reply for unexpected fd *fd***

Explanation

The SQE query engine could not write a response to the connection.

System action

Connection is closed.

Operator response

Contact system programmer.

System programmer response

Determine if the agent is available, reestablish the connection.

Module

SQEQUEUE

Procedure name

ret_response

EZA6314E **error code *err_code* : unknown request type**

Explanation

The SQE query engine received an unknown request type and could not respond.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Check the sending device and determine why it is sending incorrect request types.

Module

SQEMKREQ

Procedure name

SQEMKREQ

EZA6316E **mkPINGport : socket**

Explanation

The SQE Query engine PING routine could not establish a socket connection for a PING request.

System action

SQEPING exits.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP Programmer's Guide and Reference for more information about socket errors.

Usually this problem is caused by an authorization failure. See the information about the configure the SNMP query engine in z/OS Communications Server: IP Configuration Guide for more information about the required authorization.

Module

SQEPING

Procedure name

mkPINGport

EZA6317E **send_ping_packet: sendto**

Explanation

The SQE Query engine PING routine was unable to send a PING packet.

System action

SQEPING exits.

Operator response

Tell the system programmer about the error.

System programmer response

Verify that the specified host is connected. Verify the that the PING was requested correctly.

Module

SQEPING

Procedure name

send_PING_packet

EZA6319W **Objectid to long: obj_id**

Explanation

The object identifier was too long.

System action

MIB@DESC exits.

Operator response

Tell the system programmer about the error.

System programmer response

Correct the object identifier and resubmit. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the correct format of the object identifier in Management Information Base (MIB) Objects.

Module

MIB@DESC

Procedure name

MIB@DESC

EZA6320W **Cannot open MIB description file: *data_set***

Explanation

SNMP was unable to open the indicated MIB description data set.

System action

MIB@DESC returns control to the calling application.

Operator response

Make sure that the indicated MIB description data set is loaded and in storage accessible to the host.

System programmer response

No action is needed.

Module

MIB@DESC

Procedure name

MIB@DESC

EZA6321E ***MIB_descript_file***

Explanation

This message is displayed with message EZA6320W to indicate the current MIB file.

System action

MIB@DESC returns control to the calling application.

Operator response

Respond as indicated by message EZA6320W.

System programmer response

Respond as indicated by message EZA6320W.

Module

MIB@DESC

Procedure name

MIB@DESC

EZA6322I **Using *MIB_descript_file* as MIB description file**

Explanation

The MIB description file was opened successfully.

System action

Processing continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MIB@DESC

Procedure name

MIB@DESC

EZA6324E **unknown MIB type: *type***

Explanation

The SQE server found an unknown type field in the MIB description file.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Correct the MIB description file, the *hlq*.MIB@DESC.DATA data set, and restart SQESERV.

Module

SNMPSEND

Procedure name

read_mib_desc_tbl

EZA6325W **out of memory in read_mibdesc_tbl**

Explanation

MIB@DESC was unable to get enough storage to process the MIB description file.

System action

Processing continues.

Operator response

Tell system programmer about the error.

System programmer response

Increase the region size for the SNMPD program.

Module

MIB@DESC

Procedure name

MIB@DESC

EZA6326E **create_SNMP_port : socket *error***

Explanation

A socket() call was unsuccessful in attempting to create an SNMP port. The port could not be created.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See Appendix A, "Additional information about remote procedure call (RPC) messages," on page 1063 for more information about tcperror return codes.

Module

SNMPSEND

Procedure name

create_SNMP_port

EZA6327E **create_SNMP_port: bind *error***

Explanation

A bind() call was unsuccessful in attempting to create an SNMP port. The port could not be created.

System action

The SQESERV module ends.

Operator response

Tell the system programmer about the error.

System programmer response

See Appendix A, “Additional information about remote procedure call (RPC) messages,” on page 1063 for more information about tcperror return codes.

Module

SNMPSEND

Procedure name

create_SNMP_port

EZA6328W **parse_SNMP_packet: cannot parse packet (ps_alloc)**

Explanation

The SQE server could not parse an SNMP request because sufficient storage could not be allocated. The request is ignored.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SNMPPARS

Procedure name

parse_SNMP_packet

EZA6329W **parse_SNMP_packet: cannot parse packet (str_setup)**

Explanation

The SQE server could not parse an SNMP request because sufficient storage could not be allocated. The request is ignored.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SNMPPARS

Procedure name

parse_SNMP_packet

EZA6330W **parse_SNMP_packet: cannot parse packet (ps2pe)**

EZA6331W ***ps_erro***

Explanation

The SQE server could not parse an SNMP request because of an error converting a presentation stream to a presentation element. The request is ignored.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SNMPPARS

Procedure name

parse_SNMP_packet

EZA6332W **parse_SNMP_packet: cannot parse packet (decode)**

Explanation

The SQE server encountered an error decoding an SNMP request. The request is ignored.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Identify the agent that sent the incorrect request and correct any problems at that agent.

Module

SNMPPARS

Procedure name

parse_SNMP_packet

EZA6333W **parse_SNMP_packet: corrupted packet**

Explanation

The SNMP server found a corrupted packet. There is a problem with the agent that sent the request. The request is ignored.

System action

Processing continues with the next request.

Operator response

Identify the SNMP agent that sent the incorrect packet, and correct any errors.

System programmer response

No action is needed.

Module

SNMPPARS

Procedure name

parse_SNMP_packet

EZA6334W **snmp_parse: store_SNMP_var cannot parse packet (length1)**

Explanation

SNMPPARS could not parse a packet because the packet was longer than expected.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Determine the device sending packets with incorrect lengths. If it is SNMPD agent, verify that the packet size was specified correctly at startup.

Module

SNMPPARS

Procedure name

store_SNMP_var

EZA6335W **snmp_parse: store_SNMP_var cannot parse packet (length2)**

Explanation

SNMPPARS could not parse a packet because the packet was longer than expected.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Determine the device sending packets with incorrect lengths. If it is SNMPD agent, verify that the packet size was specified correctly at startup.

Module

SNMPPARS

Procedure name

store_SNMP_var

EZA6336W **store_SNMP_var: unknown MIB type: *type***

Explanation

The SQE server received a response containing an object with an unknown MIB type. The request is ignored. The object is not processed.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Identify the agent sending the incorrect response, and correct any problems with the agent.

Module

SNMPPARS

Procedure name

store_SNMP_var

EZA6337W **make_SNMP_packet: illegal SNMP request: *request***

Explanation

The SQE server found an incorrect SNMP request. The request is ignored.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6338W **make_SNMP_request: illegal object identifier: *id***

Explanation

The SQE server found an incorrect object identifier, which was ignored. The request is ignored.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

An application has provided an incorrect object identifier. Identify the application and correct the error. If you are using an IBM application program (for example, the NetView program), contact the IBM Software Support Center.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6339W **make_SNMP_request: out of memory (oid_cpy)**

Explanation

A call to oid_cpy was unsuccessful because it could not allocate the storage needed.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Increase the region size.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6340W **make_SNMP_packet: no MIB type defined: *name***

Explanation

An application has tried to issue an SNMP Set request for an object for which no MIB description data is known. The request is ignored.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Add the relevant data to the MIB description file and restart SQESERV. This is the *hlq*.MIB@DESC.DATA data set.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6341W **make_SNMP_request: cannot parse packet (encode)**

Explanation

A call to make_SNMP_request was unsuccessful. Preceding messages further define the nature of the problem.

System action

Processing continues.

Operator response

Tell system programmer about the problem.

System programmer response

If the problem is due to incorrect packet information or an incorrect packet format, contact the IBM Software Support Center.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6342W **make_SNMP_packet: cannot parse packet (ps_alloc)**

Explanation

The SQE server could not allocate sufficient storage to parse a packet. The request is ignored.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6343W **make_SNMP_packet: cannot parse packet (str_setup)**

Explanation

The SQE server could not parse an SNMP request because sufficient storage could not be allocated. The request is ignored.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6344W **make_SNMP_packet: cannot parse packet (pe2ps)**

Explanation

The SQE server could not parse an SNMP request because of an error converting a presentation element to a presentation stream. The request is ignored.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Invalid data was passed in the SNMP request. Check the application program that initiated the request. If you are using an IBM application program (for example, the NetView program), contact the IBM Software Support Center.

Module

SNMPPARS

Procedure name

make_SNMP_request

EZA6345W **snmp_parse: do_qb2str: out of memory**

Explanation

The SQE server could not parse an SNMP request because sufficient storage could not be allocated. The request is ignored.

System action

Processing continues.

Operator response

Tell the system programmer about the problem if it persists.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SNMPPARS

Procedure name

do_qb2str

EZA6346W **snmp_parse: is_invalid_ObjectID: illegal object identifier *obj_id***

Explanation

The SQE server encountered an incorrect object identifier. The request is ignored.

System action

Processing continues.

Operator response

Tell the system programmer about the problem.

System programmer response

Check the sending device to determine why it is sending an object identifier that is not allowed. Correct the object identifier and resubmit. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the correct format of the object identifier in "Management Information Base (MIB) Objects."

Module

SNMPPARS

Procedure name

do_qb2str

EZA6347W **out of memory in cache_object (new)**

Explanation

The SQE server could not allocate storage to add a new object to a list. No caching is done.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SQECACHE

Procedure name

cache_object

EZA6348W **out of memory in cache_object *name***

Explanation

The SQE server could not allocate enough storage to add an object to a list. No caching is done.

System action

Processing continues.

Operator response

Tell the system programmer about the error.

System programmer response

Start the SNMPQE procedure with a larger region size.

Module

SQECACHE

Procedure name

cache_object

EZA6351E **error code *err_code*: illegal protocol.**

Explanation

The SNMP query engine received a packet with an unknown protocol. The packet is discarded.

System action

The SNMP query engine continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending incorrect packets to the SNMP query engine and why it is sending them.

Module

SQEPNV

Procedure name

sqepnv

EZA6352E error code *err_code*: illegal character set

Explanation

The SNMP query engine received a packet with a character other than ASCII or EBCDIC. The packet is discarded.

System action

The SNMP query engine continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending incorrect packets to the SNMP query engine and why it is sending them.

Module

SQEPNV

Procedure name

sqepnv

EZA6353E error code *err_code*: illegal packet type

Explanation

The SNMP query engine received a packet that it did not recognize. The packet is discarded.

System action

The SNMP query engine continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending incorrect requests to the SNMP query engine and why it is sending them. See z/OS Communications Server: IP User's Guide and Commands for more information valid SNMP commands.

Module

SQEPNV

Procedure name

sqepnv

EZA6354E error code *err_code*: illegal request packet type

Explanation

The SNMP query engine received a request packet that it did not recognize. The packet is discarded.

System action

The SNMP query engine continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending incorrect requests to the SNMP query engine and why it is sending them. See z/OS Communications Server: IP User's Guide and Commands for more information valid SNMP commands.

Module

SQEPNV

Procedure name

sqepnv

EZA6355E **error code *err_code*: null get variable list**

Explanation

The SNMP query engine received a GET request that did not have any variables passed with it. The packet is discarded.

System action

The SNMP query engine continues.

Operator response

See z/OS Communications Server: IP User's Guide and Commands for more information about the valid format for SNMP GET commands. The variables expected by the SNMP query engine are the host name, community name, and the variable name(s).

System programmer response

No action is needed.

Module

SQEPNV

Procedure name

nv_get_variable

EZA6356E **error code *err_code*: unknown MIB variable**

Explanation

The SNMP query engine tried to locate the Management Information Base (MIB) variable. Either the MIB variable was unable to be located, the ANSI name is longer than 200 characters, the end of string character was not located at the end of the MIB variable, or the MIB variable type could not be located.

System action

The SNMP query engine continues.

Operator response

Verify that the *hlq*.MIBDESC.DATA is correct. See z/OS Communications Server: IP User's Guide and Commands. Verify that the request to the SNMP query engine was done correctly. Check the SNMP agent job output for messages that indicate a problem. See z/OS Communications Server: IP User's Guide and Commands for more information about the valid format for SNMP query commands and MIB variable names.

System programmer response

No action is needed.

Module

SQEPNV

Procedure name

sqepnv

EZA6357E **unrecognized type: *type***

Explanation

The SNMP query engine was unable to determine the data type within a MIB variable.

System action

The SNMP query engine stops processing the mib variable. The SNMP query engine continues.

Operator response

Verify that the *hlq*.MIBDESC.DATA is correct. See z/OS Communications Server: IP User's Guide and Commands chapter on "Managing TCP/IP Network Resources Using SNMP." Verify that the request to the SNMP query engine was done correctly. See z/OS Communications Server: IP User's Guide and Commands for more information about the valid format for SNMP query commands.

System programmer response

No action is needed.

Module

SQEPNV

Procedure name

sqepnv

EZA6358W **null hdr pointer**

Explanation

The pointer to the packet header is equal to 0.

System action

The SNMP query engine stops processing the packet and continues.

Operator response

Tell the system programmer about the error.

System programmer response

View job output for the SNMP query engine and look for any other messages that may show more information about the problem. If the problem persists contact the IBM Software Support Center.

Module

SQEPRNV

Procedure name

prNVparse

EZA6359I	**major version *major version***
EZA6360I	**minor version *minor version***
EZA6361I	**release *release***

Explanation

These messages echo the major and minor version numbers and the release levels of the received packet.

System action

The SNMP query engine processes the packet and continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVparse

EZA6362W	**illegal protocol specification**

Explanation

The SNMP query engine has received a packet with an incorrect protocol specification. The SNMP query engine stops processing the packet.

System action

The SNMP query engine continues.

Operator response

Contact the system programmer.

System programmer response

View the previous messages EZA6359 - EZA6361 for the packets major and minor versions and the release. Determine the device that is sending packets with incorrect protocol specifications and investigate.

Module

SQEPRNV

Procedure name

prNVparse

EZA6363I **native set:** ***character set***

Explanation

The SNMP query engine is displaying the native character set for the packet.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVparse

EZA6364I **packet type:** ***packet type***

Explanation

The SNMP query engine is displaying the type of packet received.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVparse

EZA6365W **packet type&colan illegal packet type (*packet type*)**

Explanation

The SNMP query engine has received an incorrect type of packet.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

If the packet was sent as a request, verify that the request was sent correctly. See z/OS Communications Server: IP User's Guide and Commands for valid SNMP commands. If the error was the result of a response for an SNMP agent, notify the system programmer.

System programmer response

Determine the SNMP agent sending incorrect responses and investigate the reason it is sending incorrect responses.

Module

SQEPRNV

Procedure name

prNVparse

EZA6366I **null request body**

Explanation

The SNMP query engine has received a request packet, but the body of the request was empty.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6367I **sequence ID: *sequence_ID***

Explanation

The SNMP query engine has received a packet that has the stated *sequence_ID*. The SNMP query engine will use the *sequence_ID* to associate requests with responses.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6368I **hostname len: *name length***

Explanation

The SNMP query engine is displaying the length of the name of the host the request will be sent to.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6369W **no hostname:**

Explanation

The SNMP query engine could not find a host name for the request packet.

System action

The SNMP query engine continues.

Operator response

Verify that the request was entered correctly.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6370I **hostname: *hostname***

Explanation

The SNMP query engine is displaying the host name that the request will be sent to.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6371I **community len: *community len***

Explanation

The SNMP query engine is displaying the length of the community name.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6372W **no community name!**

Explanation

The SNMP query engine could not find a community name for the request packet.

System action

The SNMP query engine continues.

Operator response

Verify that the request was entered correctly.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6373I **community name: *community name***

Explanation

The SNMP query engine is displaying the community name that the request was made for.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6374I **optional length *optional length***

Explanation

The SNMP query engine is displaying the number of retry parameters that SNMPIUCV has received from the SNMPARMS member of the DSIPARMS data set.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6375I **max. retries: *snmprcnt***

Explanation

The SNMP query engine is displaying the number of times that the SNMP query engine will resend an SNMP PDU when no response was received.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6376I **initial timeout *snmprito***

Explanation

The SNMP query engine is displaying the timeout value for a request (in tenths of a second). After sending an SNMP request to an agent, the SNMP query engine waits *n* tenths of a second for a response.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6377I **backoff exponent: *snmpreto***

Explanation

The SNMP query engine is displaying the retry back-off exponent. The retry back-off exponent specifies whether the timeout value between retries of an SNMP request is calculated linearly or exponentially. The valid values are 1 (linear) or 2 (exponential). The default is 2.

System action

The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVrequest

EZA6378W **illegal packet type (** ***packet type*** **)**

Explanation

The SNMP query engine has received an incorrect type of packet.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

If the packet was sent as a request, verify that the request was sent correctly. See z/OS Communications Server: IP User's Guide and Commands for valid SNMP commands. If the error was the result of a response for an SNMP agent, notify the system programmer.

System programmer response

Determine the SNMP agent sending incorrect responses and investigate the reason it is sending incorrect responses.

Module

SQEPRNV

Procedure name

prNVparse

EZA6379I **null list hdr**

Explanation

The SNMP query engine has received a request packet from SNMPIUCV that has an empty list header.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6380I	**name length *length***

EZA6381I	**name *name***

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. Displayed is the length and the name of the variable being sent or requested.

System action

The SNMP query continues to process the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6382I	**value type *type***

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. Displayed is the value type of the data being processed. The valid value types are:

text
number
internet address
empty
counter
gauge
time ticks
display string
string
object ID

System action

The SNMP query continues to process the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6383W **value type: unknown type!**

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. The value type received is not a known value type. The valid value types are:

text
number
internet address
empty
counter
gauge
time ticks
display string
string
object ID

System action

The SNMP query engine tries to process the packet. The SNMP query engine continues.

Operator response

Verify that the *hlq*.MIBDESC.DATA correct. Verify that the request to the SNMP query engine was done correctly. See z/OS Communications Server: IP User's Guide and Commands for more information about the valid format for SNMP query commands.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6384I **value length :*length***

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. The value length is being displayed.

System action

The SNMP query engine continues to process the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6385I **value :*string value***

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. The Management Information Base (MIB) variable is being displayed.

System action

The SNMP query engine continues to process the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6386I **value :*decimal value***

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. The Management Information Base (MIB) variable is being displayed.

System action

The SNMP query engine continues to process the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6387I	**value :*unsigned value***

Explanation

The SNMP query engine has received a request or is sending a response to a SNMPIUCV packet. The Management Information Base (MIB) variable is being displayed.

System action

The SNMP query engine continues to process the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6388I	**major error: *major error number***
EZA6389I	**minor error: *minor error number***
EZA6390I	**error index: *error index***
EZA6391I	**error text len: *error text len***
EZA6392I	**error text: *error text***

Explanation

The SNMP query engine has received a response to a SNMPIUCV packet. The major and minor error codes are being displayed along with any error text. For more information about SNMP error codes see z/OS Communications Server: IP User's Guide and Commands.

System action

The SNMP query engine continues to process the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVgetreq, prNVsetreq

EZA6393W **null response request body**

Explanation

The SNMP query engine has received a response packet from an SNMP agent. The body of the response was empty.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

Contact the system programmer.

System programmer response

Determine the SNMP agent that is sending incorrect packets to the SNMP query engine. Investigate why the agent is sending incorrect packets.

Module

SQEPRNV

Procedure name

prNVresp_copy

EZA6394I **filter ID:*filter id***

Explanation

The SNMP query engine has received a trap response or trap request packet. Displayed is filter ID used to filter the trap responses.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVresp_copy, prNVtrap, prNVtrap_req

EZA6395I **agent address: *agent address***

Explanation

The SNMP query engine has received a trap response or trap request packet. Displayed is the IP address of the agent sending the trap response.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVresp_copy, prNVtrap

EZA6396I **network mask *network mask***

Explanation

The SNMP query engine has received a trap response packet. Displayed is the network mask that is ANDed with the dotted decimal IP address. The result will be compared with the wanted network.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVresp_copy

EZA6397I **desired network: *network mask***

Explanation

The SNMP query engine has received a trap response packet. Displayed is the wanted network. The wanted network will be compared to the results from ANDing the agent address with the network mask.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVresp_copy

EZA6398I **null trap body**

Explanation

The SNMP query engine has received a trap packet that has no data associated with it.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVtrap

EZA6399I **generic trap: *decimal_value*(*hex_value*)**

Explanation

The SNMP query engine has received a trap packet. The generic trap type value is displayed. See z/OS Communications Server: IP User's Guide and Commands for more information about SNMP generic trap types.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVtrap

EZA6400I **specific trap: *decimal_value*(*hex_value*)**

Explanation

The SNMP query engine has received a trap packet. The specific trap type value is displayed. The specific trap value will be 0 unless the generic trap type is 6. For more information about generic and specific traps see z/OS Communications Server: IP User's Guide and Commands.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVtrap

EZA6401I **time stamp: *time stamp***

Explanation

The SNMP query engine has received a trap packet. The time stamp that was present within the trap packet is displayed.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVtrap

EZA6402I **enterprise len: *length***

EZA6403I **enterprise: *enterprise***

Explanation

The SNMP query engine has received a trap packet The length of the enterprise object ID and the enterprise object ID are displayed. See z/OS Communications Server: IP User's Guide and Commands for more information about enterprise object IDs.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVtrap

EZA6404W	**null var_name body**

Explanation

The SNMP query engine tried to parse the variable name out of the packet. The body of the packet that should contain the name was empty.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

Verify that the request was done properly and resubmit.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVvar_name

EZA6405I	**object ID len: *length***

EZA6406I	**object ID: *object id***

Explanation

The SNMP query engine has received a var_name packet. Displayed is the object ID and its length. See z/OS Communications Server: IP User's Guide and Commands. for more information about enterprise object IDs.

System action

The SNMP query engine continues processing the packet.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNvar_name

EZA6407W **null ping_req body**

Explanation

The SNMP query engine received a ping request packet. The body of the packet was empty.

System action

The SNMP query engine stops processing the packet. The SNMP query engine continues.

Operator response

Verify that the ping request was done correctly and resubmit.

System programmer response

No action is needed.

Module

SQEPRNV

Procedure name

prNVping_req

EZA6408I **STOP command received - SNMP Query Engine stops**

Explanation

The SNMP query engine received the MVS STOP command and ended.

System action

SNMP query engine ends.

Operator response

None.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: SNMP Query Engine

Module

SQESERV

Routing code

10

Descriptor code

12

Example

Not applicable.

EZA6409I ***socket_type*** **listening socket** ***socket_value*** **removed from select mask**

Explanation

The indicated listening socket has been removed from the selection mask because the maximum number of sockets are in use. When the listening socket is not included in the selection mask, new connections are not accepted for this socket.

In the message text:

socket_type
Indicates whether the listening socket type is TCP or IUCV.

socket_value
Indicates the listening socket value.

When an existing connection closes, message EZA6410I is generated in the output file and new connections can be accepted again.

System action

The SNMP query engine continues processing existing connections, but no new connections are accepted for this listening socket.

Operator response

None.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: SQESERVE

Module

SQESERV

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
EZA6290I Accepted new client connection
EZA6409I TCP listening socket 5 removed from select mask
EZA6293I Terminated client connection
EZA6410I TCP listening socket 5 added back to select mask
EZA6290I Accepted new client connection
```

EZA6410I *socket_type* listening socket *socket_value* added back to select mask

Explanation

This message is generated when the listening socket is added to the selection mask. Adding the socket to the selection mask allows new connections to be accepted for the socket.

In the message text:

socket_type
Indicates whether the listening socket type is TCP or IUCV.

socket_value
Indicates the listening socket value.

See related message EZA6409I.

System action

The SNMP query engine continues.

Operator response

None.

System programmer response

None.

User response

Not applicable.

Problem determination

Not applicable.

Source

z/OS Communications Server TCP/IP: SQESERVE

Module

SQESERV

Routing code

Not applicable.

Descriptor code

Not applicable.

Automation

Not applicable.

Example

```
EZA6290I Accepted new client connection
EZA6409I TCP listening socket 5 removed from select mask
EZA6293I Terminated client connection
EZA6410I TCP listening socket 5 added back to select mask
EZA6290I Accepted new client connection
```

EZA6423I ***mibdesc_file_name*(*line_num*), illegal object identifier :*object_id***

Explanation

The SNMP agent is checking the MIBDESC.DATA data set. Displayed is the MIBDESC data set name, line number, and the object ID that was not correct.

System action

The SNMP agent continues processing the MIBDESC file.

Operator response

Contact system programmer.

System programmer response

Correct the MIBDESC.DATA data set.

Module

MIB@DESC

Procedure name

EZA6424I **Decoded SNMP PDU :**

Explanation

SNMP has received an SNMP message.

System action

The SNMP continues processing the SNMP message.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

SNMPPRNT

Procedure name

print_SNMP_Message

EZA6425W **Message: out of memory**

Explanation

The SNMP Query Engine was decoding an SNMP message, but the SNMP Query Engine could not allocate enough storage to build the decoded SNMP message. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message, encode_SNMP_Message

EZA6426W **version: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the SNMP version number. The SNMP message is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_Message

EZA6427E **version has unknown component: *component***

Explanation

While encoding or decoding the presentation element for the SNMP version number in an SNMP request, the SNMP Query Engine was expecting the presentation element to have a version number of 0 (SNMP version-1). The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

For errors in decoding incoming SNMP requests, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors encoding an SNMP response or trap PDU to be sent out, contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message, encode_SNMP_Message

EZA6428E **Message bad sequence: *pe_error***

Explanation

If decoding the presentation element for an SNMP request, the SNMP Query Engine was expecting a presentation element with a class of Sequence to have a form of Constructed. If encoding an SNMP response or trap PDU, the SNMP Query Engine received the indicated pe_error while adding a presentation element to a sequence. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

For errors occurring on incoming SNMP messages, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors occurring on outgoing SNMP PDUs (other than out of storage errors), contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message, encode_SNMP_Message

EZA6429W **community initialization fails**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the message being encoded does not have a valid community name. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

encode_SNMP_Message

EZA6430W **community: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the community name. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_Message

EZA6431E **PDUs invalid choice selected: *choice***

Explanation

While encoding an SNMP PDU, the SNMP Query Engine was expecting the type to be 1 (get request), 2 (getNext request), 3 (get response), 4 (set request), or 5 (trap). Instead, the PDU had the indicated type. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

encode_SNMP_PDU

EZA6432W **PDU: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation elements for the PDU. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_PDU

EZA6433W **request-id: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the request-ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_PDU

EZA6434E **PDU bad sequence: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while encoding or decoding the data portion of an SNMP request. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

For errors encountered while decoding an incoming PDU, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors encountered while encoding an SNMP PDU, contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU, encode_SNMP_PDU

EZA6435W **error-status: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the error-status. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_PDU

EZA6436E **error-status has unknown component: *component***

Explanation

While encoding or decoding the error-status presentation element of an SNMP PDU, the SNMP Query Engine was expecting the error-status to be 0 (no error), 1 (too big), 2 (no such name), 3 (bad value), 4 (read only), or 5 (general error). Instead, the error-status had the indicated value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

For errors encountered while decoding an incoming PDU, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors encountered while encoding an outgoing PDU, contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU, encode_SNMP_PDU

EZA6437W **error-index: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU to be sent out, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the error-index. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_PDU

EZA6438W **Trap-PDU: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP TRAP PDU, but the SNMP Query Engine could not allocate enough memory to build the presentation element for the PDU. The TRAP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_trap_PDU

EZA6439W **enterprise initialization fails**

Explanation

The SNMP Query Engine was encoding an SNMP TRAP PDU, but the trap being encoded does not contain a valid enterprise object ID. The TRAP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

encode_SNMP_trap_PDU

EZA6440W **enterprise: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP TRAP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the enterprise object ID. The SNMP TRAP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_trap_PDU

EZA6441E **Trap-PDU bad sequence: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while encoding or decoding an SNMP TRAP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

For errors encountered while decoding an incoming TRAP, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors encountered while encoding an SNMP TRAP PDU, contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU, encode_SNMP_trap_PDU

EZA6442W **generic-trap: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP TRAP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the generic-trap type. The SNMP TRAP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_trap_PDU

EZA6443E **generic-trap has unknown component: *component***

Explanation

While encoding or decoding the presentation element for the generic-trap type in an SNMP TRAP PDU, the SNMP Query Engine was expecting the generic-trap to be 0 (coldStart), 1 (warmStart), 2 (linkDown), 3 (linkUp), 4 (authenticationFailure), 5 (egpNeighborLoss), or 6 (enterpriseSpecific). Instead, the generic-trap type had the indicated value. The SNMP trap is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

For errors encountered while decoding an incoming SNMP TRAP PDU, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that originated the trap. If the received encode trap is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors encountered while encoding an outgoing SNMP TRAP PDU, contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU, encode_SNMP_trap_PDU

EZA6444W **specific-trap: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP TRAP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the specific trap type. The SNMP TRAP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_trap_PDU

EZA6445W **VarBind: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough memory to build the presentation element for the variable bindings pair. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_VarBind

EZA6446E **VarBind bad sequence: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while encoding or decoding the variable bindings in an SNMP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

For errors encountered while decoding an incoming PDU, run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center. For errors encountered encoding the PDU, contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBind, encode_SNMP_VarBind

EZA6447W **VarBindList: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the variable bindings list. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, start the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SNMPTYPE

Procedure name

encode_SNMP_VarBindList

EZA6448E **Message bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element for the SNMP request, the SNMP Query Engine was expecting the presentation element to have a class of 0 (universal), a form of 1 (constructed), and an ID of X'10'x (sequence). Instead, the encoded request had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6449E **Message bad form: *form***

Explanation

While decoding the presentation element for an encoded SNMP message, the SNMP Query Engine was expecting the presentation element to have a form of 1 (constructed). Instead, the SNMP Query Engine request had the indicated form. The SNMP request is discarded.

This message might also appear if VTAM is closed before the SNMP Query Engine (SNMPQE) task is closed.

System action

Processing continues with the next request.

Operator response

Ensure that the SNMPQE task is closed before VTAM. If the error persists, contact the system programmer.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6450W **out of memory**

Explanation

SNMP could not allocate enough storage to decode an incoming SNMP request. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES, SNMPTYPE

Procedure name

decode_SNMP_Message, decode_SNMP_PDUs, decode_SNMP_PDU decode_SNMP_VarBindList decode_SNMP_VarBind decode_SNMP_trap_PDU decode_SMI_ObjectSyntax decode_SMI_Counter decode_SMI_Gauge decode_SMI_TimeTicks

EZA6451E **version bad class/form/id: *class/form/id***

Explanation

While decoding an incoming SNMP request, the SNMP Query Engine was expecting the presentation element for the version number to have a class of 0 (universal), a form of 0 (primitive), and an ID of 2 (integer). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6452E **version bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the SNMP version number in an SNMP request. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6453E **Message missing version element**

Explanation

While decoding an incoming SNMP request, the SNMP Query Engine could not find a presentation element for the SNMP version. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6454E **community bad class/form/id: *class/form/id***

Explanation

While decoding an incoming SNMP request, the SNMP Query Engine was expecting a presentation element for the community name to have class of 0 (universal) and an ID of 4 (octet string). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6455E **community bad octetstring: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the community name in an incoming SNMP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the pe_error indicates a storage problem, restart the SNMP Query Engine with a larger region size. Otherwise, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6456E **Message missing community element**

Explanation

While decoding an SNMP request, the SNMP Query Engine could not find a presentation element for the community name. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6457E **Message missing data element**

Explanation

While decoding an incoming SNMP request, the SNMP Query Engine could not find any presentation elements for the data. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6458E **Message has too many elements(3):** ***number***

Explanation

While decoding an SNMP request, the SNMP Query Engine was expecting the SNMP request to have 3 presentation elements (version, community name, and data). Instead, the request had the indicated number of elements. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_Message

EZA6459E **PDUs has unknown choice:** ***class/form/id***

Explanation

While decoding the presentation element of an SNMP message, the SNMP Query Engine was expecting the data portion to have a class of CONT (context-specific) and an ID of 0 (Get Request), 1 (GetNext Request), 2 (Get Response), 3 (Set Request), or 4 (trap). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDUs

EZA6460E **GetRequest-PDU bad class/id: *class/id***

Explanation

While decoding the presentation element for the data portion of the SNMP Get PDU, the SNMP Query Engine was expecting the presentation element to have a class of CONT (context-specific) and an ID of 0 (Get). Instead, the request had the indicated class and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the Recoverable error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_GetRequest_PDU

EZA6461E **GetNextRequest-PDU bad class/id: *class/id***

Explanation

While decoding the presentation element for the data portion of the SNMP GetNext Request PDU, the SNMP Query Engine was expecting the presentation element to have a class of CONT (context-specific) and an ID of 1 (GetNext). Instead, the presentation element had the indicated class and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_GetNextRequest_PDU

EZA6462E **GetResponse-PDU bad class/id: *class/id***

Explanation

While decoding the presentation element for the data portion of the SNMP GetResponse PDU, the SNMP Query Engine was expecting a class of CONT (context-specific) and an ID of 2 (Get Response). Instead, the presentation element had the indicated class and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_GetResponse_PDU

EZA6463E **SetRequest-PDU bad class/id: *class/id***

Explanation

While decoding an SNMP SetRequest, the SNMP Query Engine expected the presentation element of the data portion of the request to contain a class of CONT (context-specific) and an ID of 3 (Set Request). Instead, the presentation element had the indicated class and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_SetRequest_PDU

EZA6464E **PDU bad class/form/id: *class/form/id***

Explanation

While decoding the data portion of an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 1 (constructed), and an ID of X'10'x (sequence). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6465E **PDU bad form: *form***

Explanation

While decoding the data portion of an SNMP PDU, SNMP was expecting the presentation element to have a form of 1 (constructed). Instead, the presentation element had the indicated form. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6466E **request-id bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element containing the request-ID for the SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 2 (integer). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6467E **request-id bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the request-ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6468E **PDU missing request-id element**

Explanation

While decoding an incoming SNMP PDU, SNMP could not find the presentation element for the request-id. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6469E **error-status bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element for the error-status of an incoming SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 2 (integer). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6470E **error-status bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the error-status value in an incoming SNMP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6471E **PDU missing error-status element**

Explanation

While decoding an incoming SNMP PDU, the SNMP Query Engine could not find the presentation element for the error-status. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6472E **error-index bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element for the error-index in a received SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 2 (integer). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6473E **error-index bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the error-index value in a received SNMP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6474E **PDU missing error-index element**

Explanation

While decoding an incoming SNMP PDU, the SNMP Query Engine could not find a presentation element for the error-index. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6475E **PDU missing variable-bindings element**

Explanation

While decoding an incoming SNMP PDU, the SNMP Query Engine could not find the presentation element for the variable bindings. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6476E **PDU has too many elements(4):** ***number***

Explanation

While decoding the data portion of an SNMP request, the SNMP Query Engine was expecting the request to have 4 presentation elements (request-id, error-status, error-index, and variable bindings). Instead, the PDU had the indicated number of elements. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_PDU

EZA6477E **Trap-PDU bad class/id: *class/id***

Explanation

While decoding the presentation element of an SNMP TRAP PDU, the SNMP Query Engine was expecting the presentation element to have a class of CONT (context-specific) and an ID of 4 (trap). Instead, the presentation element had the indicated class and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6478E **enterprise bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element for the enterprise object identifier in a received TRAP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 6 (objectID). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6479E **enterprise bad object identifier: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the enterprise object identifier for a TRAP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6480E **Trap-PDU missing enterprise element**

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine could not find a presentation element for the enterprise object ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6481E **Trap-PDU missing agent-addr element**

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine could not find a presentation element for the agent address. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6482E **generic-trap bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element for the generic-trap type in an SNMP TRAP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 2 (integer). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6483E **generic-trap bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the value of the generic-trap type in an SNMP TRAP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6484E **Trap-PDU missing generic-trap element**

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine could not find a presentation element for the generic trap type. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6485E **specific-trap bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element for the specific trap type for an SNMP TRAP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 2 (integer). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6486E **specific-trap bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the value of the specific-trap type. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6487E **Trap-PDU missing specific-trap element**

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine could not find a presentation element for the specific trap type. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6488E **Trap-PDU missing time-stamp element**

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine could not find the presentation element for the time stamp. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6489E **Trap-PDU missing variable-bindings element**

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine could not find the presentation element for the variable bindings. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6490E **Trap-PDU has too many elements(6):** ***number***

Explanation

While decoding an SNMP TRAP PDU, the SNMP Query Engine was expecting the data portion of the message to have 6 presentation elements (enterprise object ID, agent address, generic trap type, specific trap type, time stamp, and variable bindings). Instead, the PDU had the indicated number of elements. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_trap_PDU

EZA6491E **VarBind bad class/form/id: *class/form/id***

Explanation

While decoding the variable bindings in an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 1 (constructed), and an ID of X'10'x (sequence). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBind

EZA6492E **VarBind bad form: *form***

Explanation

While decoding the variable bindings in an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a form of 1 (constructed). Instead, the presentation element had the indicated form. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBind

EZA6493E **VarBind missing name element**

Explanation

While decoding the variable bindings in an SNMP PDU, the SNMP Query Engine could not find the presentation element for the variable name. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBind

EZA6494E **VarBind missing value element**

Explanation

While decoding the variable bindings in an SNMP PDU, the SNMP Query Engine could not find the presentation element for the variable value. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBind

EZA6495E **VarBind has too many elements(2): *number***

Explanation

While decoding the variable bindings in an SNMP PDU, the SNMP Query Engine was expecting the variable bindings pair to have 2 presentation elements (variable name and variable value). Instead, the variable bindings had the indicated number of elements. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBind

EZA6496E **VarBindList bad class/form/id: *class/form/id***

Explanation

While decoding the variable bindings list in an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 1 (constructed), and an ID of X'10'x (sequence). Instead, the presentation element had the indicated class, form, and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBindList

EZA6497E **VarBindList bad form: *form***

Explanation

While decoding the variable bindings list in an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a form of 1 (constructed). Instead, the presentation element had the indicated form. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBindList

EZA6498E **VarBindList bad sequence: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the variable bindings list in an SNMP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SNMPTYPE

Procedure name

decode_SNMP_VarBindList

EZA6499E **ObjectName initialization fails**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the PDU being encoded does not have a valid object name. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_ObjectName

EZA6500W **ObjectName: out of memory**

Explanation

The SNMP Query Engine was encoding an SNMP PDU, but the SNMP Query Engine could not allocate enough storage to build the presentation element for the object name. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_ObjectName

EZA6501W **number: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *number*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6502E **string initialization fails**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *string*. The PDU did not have a valid string to be encoded. The PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6503W **string: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *string*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6504E **object initialization fails**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *object id*. The PDU being encoded did not have the object ID value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6505W **object: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *object id*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6506W **empty: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *empty*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6507E **ObjectSyntax invalid choice selected: *choice***

Explanation

While encoding the variable bindings of an SNMP PDU, the SNMP Query Engine expected the syntax of the variable to be 1 (number), 2 (string), 3 (object ID), 4 (empty), 5 (internet address), 6 (counter), 7 (gauge), 8 (time-ticks), or 9 (opaque). Instead, the variable had the indicated syntax. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_ObjectSyntax

EZA6510E **NetworkAddress invalid choice selected: *type***

Explanation

While encoding the agent address of a TRAP PDU, the SNMP Query Engine was expecting the network address type to be 1 (internet address). Instead, the network address had the indicated type. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_NetworkAddress

EZA6511E **IpAddress initialization fails**

Explanation

The SNMP Query Engine was encoding the agent address of an SNMP TRAP PDU or the variable bindings of an SNMP PDU. The syntax of the variable is *internet address*. The PDU being encoded did not have a valid internet address to be encoded. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_IpAddress

EZA6512W **IpAddress: out of memory**

Explanation

The SNMP Query Engine was encoding the agent address of an SNMP TRAP PDU or the variable bindings of an SNMP PDU. The syntax of the variable is *internet address*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the internet address. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_IpAddress

EZA6513W **Counter: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *counter*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_Counter

EZA6514W **Gauge: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *gauge*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_Gauge

EZA6515W **TimeTicks: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *time ticks*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_TimeTicks

EZA6516E **Opaque initialization fails**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *opaque*. The SNMP PDU being encoded did not have a valid value for the variable. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

encode_SMI_Opaque

EZA6517W **Opaque: out of memory**

Explanation

The SNMP Query Engine was encoding the variable bindings of an SNMP PDU. The syntax of the variable is *opaque*. The SNMP Query Engine could not allocate enough storage to build the presentation element for the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, restart the SNMP Query Engine with a larger region size.

System programmer response

No action is needed.

Module

SMITYPES

Procedure name

encode_SMI_Opaque

EZA6518E **ObjectName bad class/form/id: *class/form/id***

Explanation

While decoding the presentation element of the variable bindings of an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a class of UNIV (universal), a form of 0 (primitive), and an ID of 6 (object ID). Instead, the presentation element had the indicated class, form, and ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectName

EZA6519E **ObjectName bad form: *form***

Explanation

While decoding the presentation element of the variable bindings of an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have a form of 0 (primitive). Instead, the presentation element had the indicated form. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectName

EZA6520E **ObjectName bad object identifier: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the variable name of an SNMP PDU. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectName

EZA6521E **number bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the variable value of an SNMP PDU. The syntax of the variable is *number*. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectSyntax

EZA6522E **string bad octetstring: *pe_error***

Explanation

The SNMP Query Engine was decoding the variable bindings of an SNMP PDU. The syntax of the variable is *string*. The SNMP Query Engine received the indicated pe_error while decoding the variable value. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectSyntax

EZA6523E **object bad object identifier: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the variable value of an SNMP PDU. The syntax of the variable is *object id*. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectSyntax

EZA6524E **ObjectSyntax has unknown choice: *class/form/id***

Explanation

While decoding the presentation element of the variable value of an SNMP PDU, the SNMP Query Engine was expecting the presentation element to have one of the following class and ID combinations:

Class
: **ID**

APPL (application-specific)
: 0 (internet address)

APPL (application-specific)
: 1 (counter)

APPL (application-specific)
: 2 (gauge)

APPL (application-specific)
: 3 (time-ticks)

APPL (application-specific)
: 4 (opaque).

UNIV (universal)
2 (number)

UNIV (universal)
4 (string)

UNIV (universal)
5 (empty)

UNIV (universal)
6 (object ID)

Instead, the presentation element had the indicated class, form, and ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_ObjectSyntax

EZA6527E **NetworkAddress has unknown choice: *class/form/id***

Explanation

While decoding the presentation element for the agent address of the SNMP TRAP PDU, the SNMP Query Engine was expecting the presentation element to have a class of APPL (application-specific) and an ID of 0 (internet address). Instead, the presentation element had the indicated class, form, and ID. The SNMP TRAP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES, SMIPRINT

Procedure name

decode_SMI_NetworkAddress

EZA6528E **IpAddress bad class/id: *class/id***

Explanation

The SNMP Query Engine was decoding the agent address of an SNMP TRAP PDU or the variable bindings of an SNMP PDU. The syntax of the variable is *internet address*. The SNMP Query Engine was expecting the presentation element for the variable value or agent address to have a class of APPL (application-specific) and an ID of 0 (internet address). Instead, the presentation element had the indicated class and ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_IpAddress

EZA6529E **IpAddress bad octetstring: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the agent address of an SNMP TRAP PDU or the variable bindings of an SNMP PDU. The syntax of the variable is *internet address*. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_IpAddress

EZA6530E **Counter bad class/id: *class/id***

Explanation

The SNMP Query Engine was decoding the variable bindings of an SNMP PDU. The syntax of the variable is *counter*. The SNMP Query Engine was expecting the presentation element for the variable value to have a class of APPL (application-specific) and a value of 1 (counter). Instead, the presentation element had the indicated class and ID. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

If the error persists, tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_Counter

EZA6531E **Counter bad integer: *pe_error***

Explanation

The SNMP Query Engine was decoding the variable bindings of an SNMP PDU. The syntax of the variable is *counter*. The SNMP Query Engine received the indicated pe_error while decoding the variable value. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_Counter

EZA6532E **Gauge bad class/id : *class/id***

Explanation

The SNMP Query Engine was decoding the variable bindings of an SNMP PDU. The syntax of the variable is *gauge*. The SNMP Query Engine was expecting the presentation element for the variable value to have a class of APPL (application-specific) and an ID of 2 (gauge). Instead, the presentation element had the indicated class and ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_Gauge

EZA6533E **Gauge bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the variable value of an SNMP PDU. The syntax of the variable is *gauge*. The SNMP request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_Gauge

EZA6534E **TimeTicks bad class/id: *class/id***

Explanation

The SNMP Query Engine was decoding the variable bindings of an SNMP PDU. The syntax of the variable is *time ticks*. The SNMP Query Engine was expecting the presentation element for the variable value to have a class of APPL (application-specific) and an ID of 3 (time ticks). Instead, the presentation element had the indicated class and ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_TimeTicks

EZA6535E **TimeTicks bad integer: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the presentation element for the variable value of an SNMP PDU. The syntax of the variable is *time ticks*. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_TimeTicks

EZA6536E **Opaque bad class/id: *class/id***

Explanation

The SNMP Query Engine was decoding the variable bindings of an SNMP PDU. The syntax of the variable is *opaque*. The SNMP Query Engine was expecting the presentation element for the variable value to have a class of APPL (application-specific) and an ID of 4 (opaque). Instead, the presentation element had the indicated class and ID. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_Opaque

EZA6537E **Opaque bad octetstring: *pe_error***

Explanation

The SNMP Query Engine received the indicated pe_error while decoding the variable bindings of an SNMP PDU. The syntax of the variable is *opaque*. The SNMP PDU is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

See z/OS Communications Server: IP and SNA Codes for information about pe_errors. Run traces (either the SNMP Query Engine traces or some form of LAN analyzer traces) to determine whether the encoded stream was in error or if the SNMP Query Engine was in error. If the encoded stream was in error, contact the appropriate support center for the SNMP client or agent that sent the request. If the received encoded request is correct, then the SNMP Query Engine is parsing the request incorrectly. Contact the IBM Software Support Center.

Module

SMITYPES

Procedure name

decode_SMI_Opaque

EZA6617I **pDPIpacket: Major=*major version*, Minor=*minor version*, Release=*release*, Type=*type***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed are the values in the snmp_dpi_hdr. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

mkdpi

EZA6618W **pDPIpacket: No support for this level of DPI**

Explanation

The SNMP agent has received a pDPIpacket request with Major release not equal to 2 or a minor release not equal to 1 or a release version not equal to 0 or 1. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Review the minor and major version levels and the release level displayed in preceding messages. Determine the agent making the request. Determine why it is making the request with an unsupported level of DPI.

Module

MKDPI

Procedure name

mkdpi

EZA6619I **pDPIget: oid=*object id***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the object identifier being requested. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

mkdpi

EZA6620I **pDPInext: group_id=*group id*, oid=*object id***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the group identifier and object identifier being requested. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

mkdpi

EZA6621E **pDPIset: Invalid packet, integer length =*length***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the length of integer to be set. The length is not equal to 4. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Determine the agent that is sending the incorrect packet. Analyze the output to determine why it is sending incorrect packets.

Module

MKDPI

Procedure name

dpi_set_packet

EZA6622I **pDPIset: oid=*object id*, type='*type*'H, value_len= *length***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the object identifier, the SNMP_TYPE and the length of the value. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_set_packet

EZA6623I **value=*decimal_value hex_value***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the value of the object identifier in long decimal and hexadecimal formats. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_set_packet

EZA6624I **value=*decimal_value hex_value***

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the value of the object identifier in unsigned long decimal and hexadecimal formats. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_set_packet

EZA6625I **value='*hex_value*'H**

Explanation

The SNMP agent has received a pDPIpacket request. Displayed is the value of the object identifier as an octet string format. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_set_packet

EZA6626I **pDPItrap: generic=*decimal_value hex_value*, specific=*decimal_value hex_value***

Explanation

The SNMP agent has received a pDPItrap request. Displayed is the value of the traps generic and specific fields. Each field is displayed in decimal and hexadecimal formats. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPItrap command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_trap_packet

EZA6627I **pDPItrap: generic=*decimal_value hex_value*, specific=*decimal_value hex_value* enterprise=*enterprise value***

Explanation

The SNMP agent has received a pDPItrap request. Displayed is the value of the traps generic and specific fields along the enterprise value. The generic and specific fields are displayed in both decimal and hexadecimal formats. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPItrap command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_trap_packet

EZA6628I **pDPItrap: generic=*decimal_value hex_value*, specific=*decimal_value hex_value* no enterprise specified.**

Explanation

The SNMP agent has received a pDPItrap request using version 1.1 or higher. Displayed are the values of the trap's generic and specific fields. The generic and specific fields are displayed in both decimal and hexadecimal formats. No enterprise value was specified. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPItrap command.

System action

The SNMP agent continues processing the pDPIpacket.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MKDPI

Procedure name

dpi_trap_packet

EZA6630E **fDPIparse: Unknown packet_type: *type***

Explanation

While parsing a packet from an SNMP subagent over the DPI interface, SNMPD was expecting the packet type to be 1 (get), 2 (getNext), 3 (set), 4 (trap), 5 (response), or 6 (register). Instead, the packet had the indicated type. The DPI request is discarded.

System action

Processing continues with the next request.

Operator response

Tell the system programmer about the error.

System programmer response

Check the DPI subagent program for errors. If there are no apparent errors in the subagent program, contact the IBM Software Support Center. If the subagent is the SNMP subagent existing in the TCPIP module, contact the IBM Software Support Center.

Module

MKDPI

Procedure name

fDPIparse

EZA6631E **cDPIpacket: Unknown packet_type**

Explanation

The SNMP agent has received a packet that is unknown.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending the incorrect packets. View the device output to determine why it is sending incorrect packets. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

Module

MKDPI

Procedure name

cDPIpacket

EZA6632E **cDPIset: Invalid packet, integer length *length***

Explanation

The SNMP agent has received a set packet. The variable to be set is an integer type. Integer types must have a length of 4.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending the incorrect packets. View the device output to determine why it is sending incorrect packets. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIpacket command.

Module

MKDPI

Procedure name

cDPIset

EZA6633E **mkMIBquery: Unknown packet type: *packet type***

Explanation

The SNMP agent has received a mkDPIregister packet. While trying to query the Management Information Base (MIB) it determined that the packet type was unknown.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending the incorrect packets. View the device output to determine why it is sending incorrect packets. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the pDPIregister. command.

Module

MKDPI

Procedure name

mkMIBquery

EZA6634E **mkDPIset: Unknown SNMP_TYPE: *type***

Explanation

The SNMP agent has received an mkDPIset packet. The SNMP_TYPE is not supported.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending the incorrect packets. View the device output to determine why it is sending packets with incorrect SNMP_TYPEs. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the mkDPIset command.

Module

MKDPI

Procedure name

dpi_set_packet

EZA6635E **mkDPIlist: Unknown SNMP_TYPE: *type***

Explanation

The SNMP agent has received an mkDPIlist packet. The SNMP_TYPE is not supported.

System action

The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

Operator response

Contact the system programmer.

System programmer response

Determine the device that is sending the incorrect packets. View the device output to determine why it is sending packets with incorrect SNMP_TYPEs. See z/OS Communications Server: IP Programmer's Guide and Reference for more information about the mkDPIlist command.

Module

MKDPI

Procedure name

dpi_set_packet

Chapter 9. EZA8*xxxx* messages

EZA8200I **MVS TCP/IP Telnet *version level***

Explanation

You are running the indicated version of TCP/IP Telnet for MVS.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTELP

Procedure name

ClearScreen

EZA8201I **Telnet will allow you to log in to a remote host**

EZA8202I **machine. Note that the “remote” host may be the**

EZA8203I **same as your local host.**

EZA8205I **You may specify a remote host when invoking TELNET.**

EZA8206I **Otherwise, you will be prompted to enter one of the**

EZA8207I **following:**

EZA8208I **a remote host, “quit”, or “?” for assistance.**

EZA8210I **Specify the remote host either via an internet**

EZA8211I **host name, or via an internet address of the form**

EZA8212I **“nn.nn.nn.nn.”.**

EZA8214I **Once connected, follow the log in and usage procedures**

EZA8215I **of the remote host.**

EZA8217I **To invoke one of several TELNET commands, hit an**

EZA8218I **appropriate attention key:**

EZA8219I **In Transparent Mode -- PA1**

EZA8220I **In Line Mode -- any PFkey but PF 1,2,3, or 13,14,15**

EZA8221I **Then, enter any of the following commands:**

EZA8222I **Help or ? -- Receive (this) assistance**

EZA8223I	**AYT -- Are You There?**
EZA8224I	**AO -- Abort Output**
EZA8225I	**BRK -- Break**
EZA8226I	**IP -- Interrupt Process**
EZA8227I	**PA1 -- Send PA1 key; ONLY if NOT in LIne Mode**
EZA8228I	**SYNCH -- Clear data path, except for TELNET commands**
EZA8229I	**Quit -- Quit the TELNET session**
EZA8231I	**If in Line Mode, the following PF settings are in force:**
EZA8232I	**PF1 or 13 -- Retrieve previous input line**
EZA8233I	**PF2 or 14 -- Scroll halfway up**
EZA8234I	**PF3 or 15 -- Turn off display of user-line; designed**
EZA8235I	**to be used before entering password**

Explanation

These messages are displayed when Help is invoked. This is displayed by typing a question mark (?) or `Help` at the command line.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

HelpMe

EZA8246E	**Telnet terminated -- Connection closed**

Explanation

The Telnet session has ended because of a connection state type of CONNECTIONclosing, SENDINGonly, or NONEXISTENT notification has been received.

System action

The Telnet connection ends. TCPIP continues.

Operator response

Reinitiate the Telnet connection if wanted.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtEndTelnet

EZA8247E **Telnet terminated**

Explanation

The Telnet connection has been ended.

System action

The Telnet connection ends. TCPIP continues.

Operator response

Reinitiate the Telnet connection if wanted.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtEndTelnet

EZA8248E **Port *‘port number’* invalid (out-of-range or not numeric.)**

Explanation

The port number specified in the *port_number* parameter of the TELNET command is incorrect. The port parameter indicates the port number to which you want to connect on the remote host. When connecting to a non-TELNET port, the data exchange must follow the protocol recognized by the other port. The default is well-known port 23.

System action

TCPIP continues.

Operator response

Check for the correct command syntax and port number and reissue the command. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetParameters

EZA8249I **Ignoring extraneous parameters after port**

Explanation

Additional parameters have been entered after the *port* number parameter of the TELNET command. These parameters are unnecessary and are being ignored.

System action

TCPIP continues.

Operator response

Check for correct command syntax and try command again.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetParameters

EZA8250I **Missing parm after TRANSLATE**

Explanation

The *data set* name of the translation table for the TRANSLATE parameter is missing. This parameter specifies a nonstandard translation table file and can be declared as an option of the TELNET command.

System action

TCPIP continues.

Operator response

Enter the correct data set name and resubmit the command. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetParameters

EZA8251I **Invalid option *"value"* ignored**

Explanation

An incorrect option in the TELNET command has been declared. This option is being ignored.

System action

TCPIP continues.

Operator response

Check for the correct command syntax and try the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetParameters

EZA8252E ***errmsg (msgnum)***

Explanation

This message indicates the return code received after the BeginTcpIp or Handle procedure was unsuccessful. The BeginTcpIp procedure is used to inform the TCPIP address space that you want to start using its services. The Handle procedure is used to specify that you want to receive notifications in the given set. This procedure is called after BeginTcpIp and before accessing the TCPIP services.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP continues.

Operator response

Consult the system programmer.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNUTMAS

Procedure name

GetParameters

EZA8253I **Host *“host ID”* Unknown.**

Explanation

The foreign host name specified in the *foreign_host* parameter of the TELNET command is incorrect or Telnet was not able to reach the destination. This parameter specifies the name or internet address of the local or remote host.

System action

TCPIP continues.

Operator response

Check the correct command syntax and host name and try the command again. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetParameters

EZA8254I **Enter foreign host (and optional port), “quit”, or “?” for help:**

Explanation

The host parameter was not specified with the TELNET command. If you do not specify the name or internet address of the local or remote host, you are prompted for the *remote host*.

System action

TCPIP continues.

Operator response

Enter the local or remote host name and optional port number. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8255E **Cannot load translate table *data set name*; *errmsg (msgnum)***

Explanation

The data set name specified in the *data set name* option of the TRANSLATE parameter of the TELNET command is incorrect. This data set name specifies the name of a nonstandard translation table.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

The Telnet connection ends. TCPIP continues.

Operator response

Check for the correct translation data set name, reinitiate the Telnet connection and resubmit the command. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8256I **Connecting to *remote host*, port *port number***

Explanation

A connection has been initiated between the server and the indicated remote host. This message also indicates the port number for which the connection has been opened on the remote host.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8259E **Waiting for TCP connection resources. Use ATTN to quit.**

Explanation

The procedure TcpWaitOpen, which initiates a TCP connection, returned a ZEROresources return code. ZEROresources means that TCP cannot handle any more connections at this time. Since the current application does not handle RESOURCESavailable notifications and the TcpWaitOpen procedure returned a ZEROresources return code, the connection is ended. The RESOURCESavailable notifications can be declared in the Notifications record. This notification is sent only if a previous TcpOpen or TcpWaitOpen returned ZEROresources. This message is displayed with EZA8260E.

System action

The Telnet connection ends. TCPIP continues.

Operator response

Reinitiate the Telnet connection.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8260E ***errmsg (msgnum)***

Explanation

This message displays the return code received after a connection has ended. This message is displayed with EZA8259E.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

The Telnet connection ends.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8261I **Quitting...bye**

Explanation

An external interrupt notification has been received by the server to stop the Telnet connection. An EXTERNALinterrupt notification is given when a simulated external interrupt occurs in your address space.

System action

The Telnet connection ends.

Operator response

Reinitiate the Telnet connection.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8262I ***errmsg (msgnum)***

Explanation

This message indicates the return code received after the Telnet connection has ended.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

The Telnet connection ends.

Operator response

Notify the system programmer of the message.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8263E **TELNET requires a 327x-type terminal**

Explanation

The terminal type currently being used is not one supported by the Telnet protocol. The TELNET command supports IBM 3270-type display stations. Examples of supported display stations are:

IBM 3178 Display Station
IBM 3179 Display Station
IBM 3180 Display Station
IBM 3191 Display Station
IBM 3192 Display Station
IBM 3193 Display Station
IBM 3194 Display Station
IBM 3275 Display Station Model 2
IBM 3276 Control Unit Display Station Models 2, 3, and 4
IBM 3277 Display Station Model 2
IBM 3278 Display Station Models 2, 3, 4, and 5
IBM 3279 Color Display Station Models 2 and 3.

System action

The Telnet connection ends.

Operator response

Run a 3270 type terminal emulator or use a 3270 type display station.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtStartTelnetInit

EZA8265I **Using Line Mode...**

Explanation

The line mode parameter has been specified for the TELNET command. The LINEMODE parameter indicates the logon operation. In line mode, the remote host's output is displayed on the screen one line at a time, without full-screen capabilities.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtFinishTelnetInit

EZA8267I **Notes on using Telnet when in Line Mode;**

EZA8268I **- To hide Password, Hit PF3 or PF15**

EZA8269I **- To enter Telnet Command, Hit PF14-12, or PF16-24**

Explanation

These messages provide information about Telnet when the LINEMODE parameter for the TELNET command has been specified.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtFinishTelnetInit

EZA8270I **Using Transparent Mode...**

Explanation

The transparent mode of operation is the default for the TELNET command. When in transparent mode, the remote host's full-screen capabilities are functional on the local terminal.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtFinishTelnetInit

EZA8272I	**Notes on using Telnet when in Transparent Mode;**
EZA8273I	**- To enter Telnet Command, Hit PA1**

Explanation

These messages provide information about using Telnet when the transparent mode has been specified in the TELNET command.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtfinishTelnetInit

EZA8275I	****** Telnet: Internal consistency failure;**
EZA8276I	***message***

Explanation

The procedure UtMustHave, which checks that a given internal consistency condition exists, was unsuccessful. This message displays the condition associated with this procedure.

Although these messages will be displayed on the console without the message number, the following table lists the message numbers. See those messages for more information.

Description	Message number
• bad case in UtToServ.	EZA8383
• BuffEmpty: bad return code: *errmsg (msgnum)*	EZA8337
• BuffEmpty: *errmsg (msgnum)*	EZA8367
• CallToServer: Bad user command.	EZA8281
• CallTcpSend: *errmsg (msgnum)*	EZA8277
• CallToServer: Bad Tag.	EZA8282
• CallToUser: bad return code.	EZA8309
• GetNote finds user attention flag wrong.	EZA8280
• GobbleChar: urgent bytes went below 0 before DM	EZA8340
• GobbleChar: urgent bytes went below 0 before DM	EZA8341
• GobbleChar: urgent bytes went below 0 before DM	EZA8342
• GobbleChar: urgent bytes went below 0 before DM	EZA8347
• GobbleChar: urgent bytes went below 0 before DM	EZA8348
• GobbleChar: urgent bytes went below 0 before DM	EZA8349
• GobbleChar: urgent bytes went below 0 before DM	EZA8350
• GobbleChar: urgent bytes went below 0 before DM	EZA8351
• GobbleChar: urgent bytes went below 0 before DM	EZA8354
• IacNoteArrives: server sent DM after Urgentspan	EZA8307
• IssueRead: Bad Return code.	EZA8369
• MainLoop: Bad notification: *notification*	EZA8320
• NegotiateOptions: bad subnegotiation sent	EZA8295
• NewLineRead: bad line feed state.	EZA8343
• NewLineRead: bad line feed state.	EZA8344
• Pop (in UtToUser) : bad stack pointer.	EZA8335
• Received Subnegotiation on Echo option	EZA8294
• Received Subnegotiation on Transmit-binary option	EZA8297
• Received Subnegotiation on Use-Eor option	EZA8296
• Received Subnegotiation on unsupported option	EZA8298

Description	Message number
• TelnetRead: Length of subnegotiation too long	EZA8352
• TelnetRead: Expecting IACnvt.	EZA8346
• TelnetRead: Length of subnegotiation too long	EZA8355
• TelnetRead: bad subnegotiation sent	EZA8356
• TelnetRead: Bad UnderstandingState.	EZA8357
• TelnetRead: urgent bytes went below 0 before DM	EZA8384
• TryToSendData: *errmsg (msgnum)*	EZA8370
• UserCommand: Bad Telnet command.	EZA8378
• UtEndTelnet: Bad Turncode from GetNextNote.	EZA8323
• UtGetBuffSpace: Bad ReturnCode.	EZA8278
• UtGetBuffSpace: UrgentSpan is negative.	EZA8317
• UtToUser: Bad CurrentChar.	EZA8358
• UtTsInit: Bad return code from the receive.	EZA8353

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

See the message number for information on the system programmer response.

Module

TNUTMAS

Procedure name

UtMustHave

EZA8277I **CallTcpSend: *errmsg (msgnum)***

Explanation

This message is displayed after an unsuccessful attempt to start a Telnet session. This message is displayed with EZA8246E.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP continues.

Operator response

Inform the system programmer of this message.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNUTMAS

Procedure name

CallTcpSend

EZA8278I **UtGetBuffSpace: Bad ReturnCode.**

Explanation

The procedure UtGetBuffSpace, which is called when a TcpSend is unsuccessful due to insufficient buffer space, has been initialized. This procedure also receives notes from the notification list until it finds a BUFFERspaceAVAILABLE note. A BUFFERspaceAVAILABLE notification is given when space becomes available on a connection. This message is displayed with EZA8246E.

System action

The Telnet connection ends.

Operator response

Reinitiate the Telnet connection.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtGetBuffSpace

EZA8279I **Connection closed**

Explanation

The Telnet connection has ended.

System action

The Telnet connection ends.

Operator response

Reinitiate the Telnet connection if required.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtGetBuffSpace

EZA8280I **GetNote finds user attention flag wrong.**

Explanation

The procedure GetNote, which checks and processes the notification queue, found that there were no user attention notifications pending or that the queue was empty. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetNote

EZA8281I **CallToServer: Bad user command.**

Explanation

An incorrect command has been received by the server. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check for the correct function keys and commands available when using transparent mode. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

CallToServer

EZA8282I	**CallToServer: Bad Tag.**

Explanation

An incorrect command or attention key has been received by the server. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check for the correct function keys and commands available when using transparent mode. See the z/OS Communications Server: IP User's Guide and Commands for more information.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

CallToServer

EZA8283I	**in SendNegotiation**

EZA8284I	**Claim is (ord) *claim*: Option is (ord) *option***

Explanation

The procedure SendNegotiation processes option negotiations. This message indicates the command code received for the option negotiation.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

SendNegotiation

EZA8285I **in FindTermName**

Explanation

The procedure FindTermName, which processes the terminal name and sends the TERMINAL-type option subnegotiation to the client, has been initialized.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

FindTermName

EZA8286I **in SendIS**

Explanation

The SendIS procedure has been initiated. This procedure processes “terminal type is” statements from the user. The command code for IS is 0.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

SendIS

EZA8287I **termname is *terminal name* TermIndexSent is *terminal index***

Explanation

Indicates the terminal name as it was declared in the terminal-type option negotiation.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

SendIS

EZA8288I **in DONTnack**

Explanation

The procedure DONTnack, which sends a DONT negotiation acknowledgment, has been initiated. A DONT command is a denial of request to perform a specified option.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

DONTnack

EZA8289I **in WONTnack**

Explanation

The procedure WONTnack, which sends a WONT negotiation acknowledgment, has been initiated. A WONT command is a refusal to perform a specified option.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

WONTnack

EZA8290I **in BackOutOptions: backing out of state (ordinal value); *status***

Explanation

The procedure BackOutOptions, which ends any transparent mode option negotiation and resets the option negotiation status, has been initiated.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

BackOutOptions

EZA8291I **in Negotiate Options; option is (ordinal value):** ***option***

EZA8292I **tag is (ordinal value):** ***tag***

Explanation

The procedure NegotiateOptions, which handles the negotiation of Telnet options, has been initiated. This message also indicates the option negotiation command.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8293I **DONT or WONT received for unknown option**

Explanation

A DONT or WONT command has been received for an option that was not specified or is not known.

System action

TCPIP continues.

Operator response

Check for the correct Telnet option command syntax and resubmit the command. See RFC 854 for more information about option negotiations. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8294I **Received Subnegotiation on Echo option**

Explanation

A subnegotiation command has been received after an Echo option. When the Echo option is in effect, the party at the end performing the echoing is expected to transmit (echo) data characters it receives back to the sender of the data character. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8295I **NegotiateOptions: bad subnegotiation sent**

Explanation

The procedure NegotiateOptions, which handles the negotiation of Telnet options, received an incorrect subnegotiation command. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check for the correct Telnet option command syntax and resubmit the command. See RFC 854 for more information about option negotiations. See Appendix B, "Related protocol specifications," on page 1065 for information about accessing RFCs.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8296I **Received Subnegotiation on Use-Eor option**

Explanation

A subnegotiation option command has been received on a Use-Eor option. End of record (EOR) is used to frame the end of a write command when in transparent mode. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8297I **Received Subnegotiation on Transmit-binary option**

Explanation

A subnegotiation option has been received on a transmit binary option. The transmit binary option allows for a transmission change to 8-bit binary. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8298I **Received Subnegotiation on unsupported option**

Explanation

The option for which a subnegotiation has been received is not implemented on this Telnet connection. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check for the correct Telnet option command syntax and resubmit the command. See RFC 854 for more information about option negotiations. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

NegotiateOptions

EZA8305I **in IacNoteArrives**

Explanation

The procedure IacNoteArrives, which handles Interpret As Command (IAC) commands received from the Telnet server, has been initiated.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

IacNoteArrives

EZA8306I **Option neg. stuff arrives**

Explanation

The procedure IacNoteArrives, which handles Interpret As Command (IAC) commands received from the Telnet server, has received a Telnet option negotiation command.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

IacNoteArrives

EZA8307I **IacNoteArrives: server sent DM after Urgentspan**

Explanation

A data mark (DM) has been received. A data mark is a reserved octet that is appended to a control function. While in Urgent mode, this option will end a program running on the remote host. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

IacNoteArrives

EZA8308I **Other Telnet char arrives**

Explanation

Additional Telnet characters are being received by the server.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

IacNoteArrives

EZA8309I **CallToUser: bad return code.**

Explanation

The procedure CallToUser, which processes return codes received from UtToUser, was unsuccessful because of an incorrect return code received. The procedure UtToUser contains the routines for correspondence from server Telnet to the user. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

This message should be preceded by more specific error messages. Correct the errors indicated by the preceding messages.

Module

TNUTMAS

Procedure name

CallToUser

EZA8310I **DataDelivered; # bytes: *bytes***

Explanation

Indicates the number of bytes delivered from the user to the Telnet server.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8311I **UrgentMode;LastUrgByte *byte***

Explanation

The server has received an attention signal to begin data transmission in urgent mode. When in urgent mode, the data stream is immediately scanned for attention signals. These signals are used to invoke special handling of the data stream by the process that receives it. This message also indicates the last urgent byte being delivered.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8312I **got USERwantsATTENTION**

Explanation

The user has entered an attention key. An attention key is a function key on terminals that, when pressed, causes an I/O interruption in the processing unit.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8313I **got USERdeliveresLINE**

Explanation

The user has entered a line of data at the terminal.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8314I **got CONNECTIONstate**

Explanation

The client has received a connection state change notification. This notification is given when a TCP connection receives information from the network that causes the state of the connection to change. Such changes might arise when the remote client opens, closes, or abends the connection.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8315I **Connection closed**

Explanation

The Telnet connection has ended.

System action

The Telnet connection ends.

Operator response

Reinitiate the Telnet connection if required.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8316I **in URGENTpending**

Explanation

An URGENTpending notification has been received. This notification is given when the TCPIP service is informed by the remote site that there is urgent data not yet delivered to the client.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8317I **UtGetBuffSpace: UrgentSpan is negative.**

Explanation

No more urgent data is found in the data stream before a data mark (DM) is received. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8318I **UrgentMode**

Explanation

Urgent mode has been initiated. When in urgent mode, the data stream is immediately scanned for attention signals. These signals are used to invoke special handling of the data stream by the process that receives it.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8319I **got EXTERNALinterrupt: *code***

Explanation

An EXTERNAL interrupt notification has been received. This notification is given when a simulated external interrupt occurs in your address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8320I **MainLoop: Bad notification: *notification***

Explanation

The procedure MainLoop, which processes data from the user to the Telnet server, was unsuccessful because of an incorrect notification received. This message displays the notification name. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

This message should be preceded by more specific error messages.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

MainLoop

EZA8321I **in UtEndTelnet**

Explanation

The procedure UtEndTelnet, which ends the user‘s Telnet connection, turns off the full-screen services, and shuts down the TCP connection, has been initiated.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtEndTelnet

EZA8322I **Session ended. <ENTER> to return to TSO**

Explanation

The Telnet connection has ended. To return to the time sharing option (TSO) environment, press the ENTER key.

System action

The Telnet connection ends. TCPIP continues.

Operator response

Press the ENTER key.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtEndTelnet

EZA8323I **UtEndTelnet: Bad TurnCode from GetNextNote.**

Explanation

The UtEndTelnet procedure, which stops a Telnet connection, received an incorrect return code from GetNextNote. The GetNextNote procedure gets the next notification off the notification queue. This message is displayed with EZA8246E.

System action

The Telnet connection is ended. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtEndTelnet

EZA8324I	**Using TCPIPTES...**

Explanation

The test option has been specified for the NETSTAT command. This command provides information about the status of the local host. The TEST option displays detailed information about the TCPIPTES address space. You can use the TEST parameter in conjunction with any other NETSTAT parameter to display information about the specified TCPIPTES address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

GetParameters

EZA8325I	**Option (ordof enum):**
EZA8326I	**WILL.**
EZA8327I	**WILL consummated.**
EZA8328I	**DO.**
EZA8329I	**DO consummated.**

Explanation

Indicates the options that are currently in effect. A WILL option indicates the need to begin performing, or confirmation that you are now performing, the indicated option. A DO option indicates the request that the other party perform, or confirmation that you are expecting the other party to perform, the indicated option. Depending on the options specified only some of these options will be displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtFinishTelnetInit

EZA8331I **Telnet**

Explanation

The Telnet services have been invoked.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNUTMAS

Procedure name

UtFinishTelnetInit

EZA8335I **Pop (in UtToUser): bad stack pointer.**

Explanation

Telnet issues this message while attempting to process data received by the server. The submitted data has been processed and no more procedure calls are required. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

Push

EZA8337I **BuffEmpty : bad return code : *errmsg (msgnum)***

Explanation

Telnet issues this message when an unsuccessful return code is given while attempting to retrieve data from the buffer for processing. This message is displayed with EZA8246E.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNTOUSE

Procedure name

GetMoreData

EZA8338I **ord: *character* asis: *character* severity. Informational.**

Explanation

This message occurs as a result of the most recent character to be processed being retrieved from the buffer, processed, and printed to the user screen.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

IacScan

EZA8339I **ord: Transparent mode, found IAC at IacOffset *number* is *character***

Explanation

Telnet issues this message as a result of an Interpret As Command (IAC) character being detected while tracing is on in transparent mode. The IAC is displayed with a command code. If this command deals with option negotiation, the command will have a number to show the code for the referenced option.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

IacScan

EZA8340I **GobbleChar: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in transparent mode. The data submitted containing urgent data bytes has been exhausted before the data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

IacScan

EZA8341I **GobbleChar: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in Line (non-transparent) mode. The data submitted containing urgent data bytes have been exhausted before detecting a data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

IacScan

EZA8342I **GobbleChar: urgent bytes went below before DM**

Explanation

Telnet issues this message while in line (non-transparent) mode. The data submitted containing urgent data bytes have been exhausted before detecting a data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

IacScan

EZA8343I **NewLineRead: bad line feed state.**

Explanation

Telnet issues this message while in line (non-transparent) mode. An incorrect line feed state was recognized while attempting to read the line feed set sequence. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Submit a valid LfSet sequence in the data set to declare a new line.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

CheckForNeedToSuspend.

EZA8344I **NewLineRead: bad line feed state.**

Explanation

Telnet issues this message while in line (non-transparent) mode. An incorrect line feed state was recognized while attempting to read the carriage return set sequence. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Submit a valid CrSet sequence in the data set to declare a new line.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

CheckForNeedToSuspend

EZA8345I **in TelnetRead**

Explanation

Telnet issues this message while tracing is on. The Telnet command, which was indicated by the presence of an Interpret As Command (IAC) character, is now being read.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8346I **TelnetRead: Excepting IACnvt.**

Explanation

Telnet issues this message while tracing is on. The client is now scanning for an Interpret As Command (IAC) character. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8347I **TelnetRead: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while tracing is on. The data submitted containing urgent bytes have been exhausted before the data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8348I **TelnetRead: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in urgent mode. While looking for a command following the IAC (Interpret As Command) character, the data containing urgent bytes has been exhausted before the data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8349I **TelnetRead: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in urgent mode. While scanning for the negotiation option following the Interpret As Command (IAC) character, the data containing urgent bytes was exhausted before detecting a data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8350I **TelenetRead: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in urgent mode. The data containing urgent bytes was exhausted before detecting a data mark while negotiating options. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8351I **TelnetRead: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in urgent mode. While scanning for a negotiation option following the Interpret As Command (IAC) character, the data containing urgent bytes was exhausted before detecting a data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8352I **TelnetRead: Length of subnegotiation too long**

Explanation

Telnet issues this message to indicate that too many negotiation commands were submitted during the negotiation process. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Resubmit the negotiation process.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8353I **TelnetRead: urgent bytes went below 0 before DM**

Explanation

Telnet issues this message while in urgent mode. While scanning for the end of the subnegotiations, the data containing urgent bytes was exhausted before detecting a data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8354I **TelnetRead: urgent byes went below 0 before DM**

Explanation

Telnet issues this message while in urgent mode. While scanning for the end of the Telnet command, the data containing urgent bytes was exhausted before detecting a data mark. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8355I **TelnetRead: Length of subnegotiation too long**

Explanation

Telnet issues this message while scanning for the end of the Telnet subnegotiations. Instead of a subnegotiation end command, an Interpret As Command (IAC) character was detected. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Resubmit the Telnet command subnegtiations.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8356I **TelnetRead: Bad subnegotiation sent.**

Explanation

Telnet issues this message to indicate that it has received an incorrect subnegotiation packet from the other side of the connection This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Resubmit the Telnet command.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8357I **TelnetRead: Bad UnderstandingState.**

Explanation

The Telnet server has received an incorrect message while expecting either subnegotiation commands or Interpret As Command (IAC) character. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check client machine for the syntax of the command being submitted.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8358I **UtToUser: Bad CurrentChar.**

Explanation

This message indicates an internal consistency error. For more information, see message EZA8275I

System action

TCPIP continues.

Operator response

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

TelnetRead

EZA8359I **Data received from TCP:**

Explanation

Telnet issues this message when more data is sent to the client machine due to data exhaustion of the buffer.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

UtToUser

EZA8361I ***address***

Explanation

Telnet issues this message while tracing is on and in transparent mode. The address of the current character is displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

UtToUser

EZA8364I ***address***

Explanation

Telnet issues this message while tracing is on and in transparent mode. The address of the current character is displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

UtToUser

EZA8366I **bad case in UtToUser**

Explanation

The command called to be processed by the Utouser procedure was not recognized.

System action

TCPIP continues.

Operator response

Check the Telnet client machine for the syntax of the command being submitted.

System programmer response

No action is needed.

Module

TNTOUSE

Procedure name

UtToUser

EZA8367I **BuffEmpty: *errmsg (msgnum)***

Explanation

Telnet issues this message while attempting to receive more data from the server. The buffer that contained the data was empty. This message is displayed with EZA8246E.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNTOUSE

Procedure name

UtTuInit

EZA8368I **Valid telnet cmds: AO,AYT,BRK,IP,PA1,QUIT,SYNCH.**

Explanation

Telnet issues this message when an incorrect Telnet subcommand is submitted. A list of valid Telnet command is displayed.

System action

TCPIP continues.

Operator response

Resubmit the Telnet command.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

FsHelpMe

EZA8369I **IssueRead: Bad Return Code.**

Explanation

Telnet attempted to create a new buffer for user input, but one already exists. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

IssueRead

EZA8370I **TryToSendData: *errmsg (msgnum)***

Explanation

An attempt to send data was unsuccessful. The reason for the unsuccessful attempt is indicated by the SayCalRe portion of this message. This message is displayed with EZA8246E.

errmsg is the text of the message that describes the error.

msgnum is the 4-digit numeric portion of the message identifier of the **EZA** message whose text is displayed in *errmsg*. For more information about this message, see message EZA*msgnum*.

System action

TCPIP continues

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*msgnum*.

Module

TNTOSRV

Procedure name

TryToSend

EZA8371I	in Senddata

Explanation

Telnet issues this message while in debug mode to indicate that it is using the procedure Senddata, which sends data to Server Telnet.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

SendData

EZA8372I	Telnet command is
EZA8373I	***command***

Explanation

Telnet issues the message while tracing is on. The current Telnet command is displayed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

Fs2ndFieldAddr

EZA8375I **‘?’ for help: Hit Enter to continue..**

Explanation

Telnet issues this message in line (non-transparent) mode to indicate that the previous command submitted was incorrect.

System action

TCPIP continues.

Operator response

Submit a correct command or type ? for help.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UserCommand

EZA8376W **AMBIGUOUS command: Hit Enter to continue..**

Explanation

Telnet was unable to determine which command was indicated by a user response due to truncation or incorrect syntax.

System action

TCPIP continues.

Operator response

Check the syntax and resubmit the Telnet command.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UserCommand

EZA8377W **Command illegal in Line-Mode: Hit Enter to continue..**

Explanation

Telnet issues this message when the PA1 key is pressed while in line mode. This key is incorrect in line (non-transparent) mode.

System action

TCPIP continues.

Operator response

Press Enter to continue.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UserCommand

EZA8378I **UserCommand: Bad Telnet Command.**

Explanation

You submitted an incorrect Telnet command. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check the syntax and resubmit the command.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UserCommand

EZA8379I **Telnet command:**

Explanation

This message prompts the user for a Telnet command. A valid Telnet command is expected.

System action

TCPIP continues.

Operator response

Enter the appropriate Telnet command.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UserDemandsAttention

EZA8380I **User data is...**

EZA8381I ***data set data set***

EZA8382I **; Len is *data set length***

Explanation

Telnet issues these messages while tracing is on. When the user data is received by the server, the address and length of the data buffer containing that information is recorded.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UserDataArrives

EZA8383I **bad case in UtToServ.**

Explanation

The command to be processed by the UtToServ procedure was not recognized. This message is displayed with EZA8246E.

System action

TCPIP continues.

Operator response

Check the Telnet server machine for the syntax of the command being submitted.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UtToServ

EZA8384I **UtTsInit: Bad return code from the receive.**

Explanation

Telnet issues this message as a result of a return code error while attempting to ready a buffer for user input.

System action

TCPIP continues.

Operator response

Increase the buffer size by using the DATABUFFERPOOLSIZE statement described in the z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

UtTsInit

EZA8395I **Once connected, follow the log in and usage**

EZA8396I **procedures of the remote host.**

EZA8397I **To invoke one of several TELNET commands, hit a**

EZA8398I **PF key (PF4-12), PF(16-24), and then enter**

EZA8399I **the following commands:**

EZA8400I **Help or ? -- Receive (this) assistance**

EZA8401I **AYT -- Are You There?**

EZA8402I **AO -- Abort Output**

EZA8403I	**BRK -- Break**
EZA8404I	**IP -- Interrupt Process**
EZA8405I	**SYNCH -- Clear data path, except for TELNET commands**
EZA8406I	**Quit -- Quit the TELNET session**
EZA8408I	**The following PF settings are in force:**
EZA8409I	**% PF1 or 13 -- Retrieve previous input line**
EZA8410I	**% PF2 or 14 -- Scroll halfway up**
EZA8411I	**% PF3 or 15 -- turn off display of user-line; designed.**
EZA8412I	**to be used before entering password**
EZA8414I	**For control characters, enter ¢c or`c where c is:**
EZA8415I	**“0”: 00, “a” - “z” or “A” - “Z”: 0x01-0x1A**
EZA8416I	**“2” - “6”: 0x1B-0x1F**
EZA8417I	**“{”: 0x5B, “}”: 0x5D, “#”: 0x7**

Explanation

Telnet issues these messages when the help (?) option is selected in line (non-transparent) mode.

System action

TCPIP continues.

Operator response

Proceed as instructed.

System programmer response

No action is needed.

Module

TNTOSRV

Procedure name

FsHelpMe

EZA8500S	**GetVirtualAddress call should not occur.**

Explanation

TCPIP has called the procedure GetVirtualAddress, which should not be called for MVS. This indicates a software error.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Contact the IBM Software Support Center.

Module

CMMVSUB

Procedure name

GetVirtualAddress

EZA8501I **Dataset: *data_set_name***

Explanation

This message displays the name of a partitioned data set.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMVSUB

Procedure name

PrintSpecOfFile

EZA8502I **Dataset: *partitioned_data_set(member_name)***

Explanation

This message displays the name of a partitioned data set and one of its members.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMVSUB

Procedure name

PrintSpecOfFile

EZA8503E ***error_message***

Explanation

This diagnostic message is preceded by an SMTP error message. This message displays the text of the Dynamic Allocation (SVC 99 - DYNALLOC) message. This message provides additional information about why the Dynamic Allocation failed. The message might be issued multiple times to include the completer text from Dynamic Allocation.

error_message is text of the message produced by DYNALLOC.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

See the z/OS MVS Programming: Authorized Assembler Services Guide for information about the IEFDB476 service routine message displayed in this message and respond as indicated.

Module

CMMVSUB

Procedure name

S99Error

EZA8520I **OK**

Explanation

The specified application or function is OK in TCPIP. Another message should prefix this message specifying the application or function.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8521I **Abnormal condition due to CSA storage shortage**

Explanation

There is no CSA storage available.

System action

TCPIP continues.

Operator response

Wait a few minutes and try again. If the problem persists, contact the System Programmer.

System programmer response

Verify the configuration and storage definitions.

Module

various.

Procedure name

various.

EZA8522I **Connection already closing**

Explanation

The connection that you are attempting to disconnect is already in the process of closing.

System action

TCPIP continues.

Operator response

If you do not want the connection to close, reestablish your connection.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8523I **Invalid length specified**

Explanation

A parameter was of an incorrect length for the remote site.

System action

TCPIP continues.

Operator response

Reissue your command with a valid parameter length specified. This error should never occur for SMTP or the FTP Server. If it does occur, this indicates that there might be an error with SMTP or the FTP Server code. If this is the case, refer the error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the SMTP or the FTP Server code with trace on for the specific application where the error occurred. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application through TCPIP.

Module

various.

Procedure name

various.

EZA8524I **Cannot send data**

Explanation

The FTP Server did not acknowledge receipt of the data sent. In the case of Telnet, this message indicates that the Telnet server could not send the function StKillConn, which closes connections, to the connection indicated in a previous message. A message following this one indicates that there is no such connection and shutdown continues.

System action

TCPIP continues. The calling program continues.

Operator response

This error should never occur in the case of FTP. If it does, this indicates that there might be an error with the FTP Server FTP code. If this is the case, refer the error to the system programmer or the IBM Software Support Center. In the case of the Telnet server, check the connection number indicated in the previous message to determine the problem. If the connection indicated is OK then the problem could be with the server code. If so, refer this problem to the IBM Software Support Center.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8525I **Client reinitialized TCP/IP service**

Explanation

A client has reinitialized TCPIP.

System action

TCPIP restarts.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8526I **Connection already exists**

Explanation

A connection to the server already exists. The request is ignored by the server.

System action

TCPIP continues. The calling program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8527I **Destination address is unreachable**

Explanation

The address that you specified is unreachable. This occurs as a result of errors from the remote site. Connection to the remote site is not established.

System action

TCPIP continues.

Operator response

Contact the remote site administrator for assistance.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8528I **Error in profile data set. Details are in PROFILE.TCPERROR or the // SYSERROR DD file. Expansion may be necessary.**

Explanation

There is an error in the *hlq.*.PROFILE.TCPIP data set.

System action

TCPIP continues.

Operator response

Open the *hlq.*.PROFILE.TCPERROR data set, or the data set pointed to by the //SYSERROR DD statement in the TCPIP JCL procedure. Respond to the details provided in the data set.

System programmer response

Assist the operator as required.

Module

various

Procedure name

various

EZA8529I **Fatal inter-address communications error VMCF RC=*number* User=*lastuser***

Explanation

The Virtual Machine Communication Facility "VMCF" encountered a network pacing response code. The VMCF return codes below describe the different types of error conditions that may cause the fatal inter-address communication error.

Return Code	Cause
1	Invalid virtual buffer address
2	Invalid subfunction code
3	Protocol violation
4	Source machine not authorized
5	TCPjobname user not available
6	Protection violation
7	Sendx data too large
9	Target machine in quiesce state
10	Message limit exceeded
11	REPLY canceled
12	Message not found
13	Synchronization error
15	Paging I/O error
16	Incorrect length
17	Destructive overlap
18	User not authorized for PRIORITY message.
19	Data transfer error
20	CANCEL - busy

System action

TCPIP halts.

Operator response

Tell the system programmer the VMCF communication error encountered.

System programmer response

Use the return code displayed in the list of error responses to determine the cause of the error.

If the return code=5, verify that the TCPIPJOBNAME is correctly specified in the TCPIP.DATA data set. Ensure that you are using the correct TCPIP.DATA data set.

For other return codes, correct the error using the different resources stated below that may apply to the error condition specified. Once the error is corrected restart TCP/IP. For more information on the VMCF facility and return codes see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference. For information on configuring the TCPIP.DATA for the VMCF, see the z/OS Communications Server: IP Configuration Reference. If the error is beyond the scope of these resources, contact the IBM Software Support Center.

Module

various

Procedure name

various

EZA8530I **No password in RACF directory**

Explanation

The password entered does not exist in the RACF directory.

System action

TCPIP continues. Connection to the remote host ends.

Operator response

Check the password and syntax and make sure that it is a valid password and try your request again.

System programmer response

Help the user update their password in the RACF directory if required.

Module

various.

Procedure name

various.

EZA8531I **TCPIP not authorized to access dataset**

Explanation

The specific data set requested through TCPIP is not accessible; the user is not authorized.

System action

TCPIP continues. The application continues.

Operator response

To access the data set you must be authorized by the remote site. Check with your system administrator to authorize your user ID to have access to the data set on the FTP server.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8532I **Invalid request**

Explanation

The FTP server has received an incorrect request from TCPIP.

System action

TCPIP halts. The calling program halts.

Operator response

This error should never occur. If it occurs it indicates an error with the server FTP code itself. Refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the specific application where the error occurred. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application through TCPIP.

Module

various.

Procedure name

various.

EZA8534I **Invalid virtual address**

Explanation

The virtual address specified to TCPIP is not valid. No connection is made to the remote host.

System action

TCPIP continues.

Operator response

Check to make sure that you entered a valid virtual address defined in the remote system. Call the remote system administrator for assistance.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8535I **You aborted the connection**

Explanation

You either intentionally or unintentionally abended the connection to the remote host.

System action

TCPIP continues. The remote host connection is abended.

Operator response

Reinitialize the connection to the remote host.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8536I **The requested local port is not available**

Explanation

The local port on which TCPIP has been accessed is not available.

System action

TCPIP halts.

Operator response

Wait several minutes or until TCPIP has been restarted. You should receive a second message after attempting to use TCPIP that reads, `TCPIP has been restarted...for some reason or other`, when TCPIP becomes available.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8537I **Data set is in use by someone else and cannot be accessed**

Explanation

The data set requested through TCPIP is in use by someone else and is not accessible at this time.

System action

TCPIP continues. The calling program continues.

Operator response

Wait a few minutes until the data set becomes available and reissue your request.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8538I **Data set not available**

Explanation

The data set requested is not available. It is either in use or inaccessible.

System action

TCPIP continues. The calling program continues.

Operator response

Wait a reasonable amount of time and try accessing the data set again. If you are not successful after several tries, contact the system programmer or the IBM Software Support Center.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8540I **The foreign host has closed this connection**

Explanation

The remote host has closed this connection.

System action

TCPIP continues. The host is unsuccessful.

Operator response

Wait until the remote host is ready and reissue the connection to the remote host.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8541I **The internet address is not local to this host**

Explanation

The host address entered though TCPIP is not local to this host. This host address verification is part of the virtual machine communication facility (VMCF) used within the MVS system. Connection to the remote host is unsuccessful.

System action

TCPIP continues.

Operator response

Check the host address entered and verify that it is correct. If it is, then it has to be updated in your VMCF data set.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8542I **No outstanding notifications**

Explanation

There is an error in the application program code itself. The application will halt. The application could include, but is not limited to; SMTP, FTP server, LPD.

System action

TCPIP continues. The calling program halts.

Operator response

Reinitialize your connection and the application program. If the problem persists, contact the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8543I **No such connection**

Explanation

The connection you attempted to query or use does not exist. For the Telnet server, this indicates that the function StKillConn which was called to close a connection was unsuccessful because the connection could not be found.

System action

TCPIP continues. In the case of the Telnet server, this message is ignored and shutdown procedures continue.

Operator response

In most cases you can try your request again using the correct spelling and syntax for the remote address to correct the error. For the Telnet server see other messages that appear with this one for more information. If the error persists, contact the IBM Software Support Center.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8544I **No TCP/IP service available**

Explanation

TCPIP is not available.

System action

TCPIP halts.

Operator response

Wait until TCPIP becomes available and try your request again.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8545I **Not yet begun TCP/IP service**

Explanation

TCPIP is not available.

System action

TCPIP halts.

Operator response

Wait until TCPIP resumes service or reinitialize TCPIP.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8546I **The connection is not yet open**

Explanation

This error should never occur. If it does it means that there is an error with the specific application code itself. Affected applications include, but are not limited to, SMTP and FTP server.

System action

TCPIP continues. The calling program continues.

Operator response

Report this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8547I **Foreign host rejected the open attempt**

Explanation

The remote host that you attempted to connect to has rejected the open attempt.

System action

TCPIP continues. The connection to the remote host is abended.

Operator response

This error occurs on the remote site. Contact the site and reissue your open attempt when their system clears up.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8548I **TcpOpen error: invalid local address**

Explanation

The local address that you attempted to use for a communication channel is not valid. The attempted connection is not established.

System action

TCPIP continues.

Operator response

This error should not occur. If the error does occur, it implies an error with the application code itself. Refer the error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program. Check the address and syntax and reissue your connection attempt. If the problem persists, contact your system administrator to verify the local address or the IBM Software Support Center for help in debugging the code.

Module

various.

Procedure name

various.

EZA8549I **TcpOpen error: invalid initial state**

Explanation

You need to specify whether you are initiating a TCPIP connection in a passive or active state. It could also indicate that there is an error with the application (FTP server, SMTP) code itself.

System action

The connection is not established.

Operator response

Reissue the connection indicating whether it will be a passive or active connection. If the problem exists in the application code itself, refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8550I **Invalid timeout parameter**

Explanation

The timeout parameter is used for the PING function when accessing a remote host. PING will return with a message that states that the remote host is not responding if after a few moments it receives no responses from the remote host. In the case of FTP server and SMTP, this could indicate an error with the application code itself.

System action

TCPIP continues. The connection is not established.

Operator response

If the error occurred as result of the PING function, then wait until the host responds or call the remote host's system administrator to find out the status of the host. In the case of FTP server and SMTP, refer the error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8551I **TcpOpen error: unspecified foreign address in active open**

Explanation

Indicates, in the case of the TCP engine, that the host address has not been specified in your open attempt. In the case of FTP and SMTP, this error indicates that there might be a problem with the application code itself. The connection is not established.

System action

TCPIP continues.

Operator response

Specify a valid host address when issuing a socket request. If the error persists, verify the address with the other host's administrator. If the error occurred within the FTP or SMTP function, refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8552I TcpOpen error: unspecified foreign port in active open

Explanation

The host port has not been specified in your open attempt. In the case of FTP and SMTP, this error indicates that there might be a problem with the application code itself.

System action

TCPIP continues. The program halts.

Operator response

Specify a valid host port when issuing a socket request. If the error persists, verify the address with the other host's administrator. If the error occurred within the FTP or SMTP function, refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8553I TCPIP cannot read profile data set

Explanation

TCPIP cannot access or read your *hlq*.PROFILE.TCPIP data set.

System action

TCPIP continues.

Operator response

Make sure that your *hlq*.PROFILE.TCPIP data set is accessible by TCPIP. If it is and you are still experiencing this error, open the data set and check the contents to make sure that the information is correct and complete and that the information in the data set is not damaged. For more information about the *hlq*.PROFILE.TCPIP data set, see z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8554I **Receive still pending on this connection**

Explanation

No data has been received on the specified connection. In the case of the FTP server and SMTP, this message could indicate a problem with the code itself.

System action

TCPIP continues.

Operator response

If the error specified is from any application other than the FTP server or SMTP, wait for data or ask the remote site to send you an acknowledgment packet to test your connection. If the error is the SMTP or the FTP server code, refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8555I **Foreign host unexpectedly closed the connection**

Explanation

The remote host has closed the connection, for the specified application, without notice or acknowledgment.

System action

TCPIP continues. The program halts.

Operator response

This message occurs as a result of a problem with the remote host. Reinitialize your connection or call the system administrator for the remote host for assistance.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8556I **Foreign host aborted the connection**

Explanation

The other host has abended the connection. This could indicate a connection reset at the other host. This could also apply to your Offload device.

System action

TCPIP continues. The program halts or is disconnected.

Operator response

In the case of the Offload device, reinitialize the connection from the Offload device. In any other application, contact the other host's administrator to reset their host or determine if there are other problems on their end causing this error.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8557I **Software error in TCP/IP!**

Explanation

TCPIP cannot continue because of an error in TCPIP itself.

System action

TCPIP halts.

Operator response

Wait a few moments until TCPIP restarts. If after a reasonable amount of time TCPIP does not restart, contact your system administrator.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8558I **TCP/IP service is being shut down**

Explanation

TCPIP is being shut down either manually or due to a problem with the application connection.

System action

TCPIP halts.

Operator response

Wait a few moments until TCPIP restarts. If after a reasonable amount of time TCPIP does not restart, contact your system administrator.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8559I **Foreign host is no longer responding**

Explanation

The connection to the other host has not responded within a certain time threshold.

System action

TCPIP continues. The connection to the remote host abends.

Operator response

Wait a few moments until TCPIP restarts. If after a reasonable amount of time TCPIP does not restart, contact the other host's system administrator.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8560I **Foreign host did not respond within OPEN timeout**

Explanation

The open attempt to the other host has not responded within a certain time threshold. The connection to the remote host is not established.

System action

TCPIP continues.

Operator response

Wait a few moments until TCPIP restarts. If after a reasonable amount of time TCPIP does not restart, contact the other host's system administrator.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8561I **Too many open connections already exist**

Explanation

The host connection that you are attempting to establish a connection with already has too many open connections. The connection is not established.

System action

TCPIP continues.

Operator response

Contact the other host's administrator to determine the availability of their system.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8563I **Permission is denied**

Explanation

The error might be caused by one of the following reasons:

- Not being APF-authorized to use the RAW IP interface
- A Firewall access failure
- A network access failure

System action

The command is not performed and TCPIP continues.

Operator response

Contact your system administrator to gain access to the specific application that you are attempting use.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8564I **Foreign host violated the connection protocol**

Explanation

The other host has sent data that does not conform to the connection protocol.

System action

TCPIP continues. The program halts or the connection is abended.

Operator response

This error arises as a problem at the remote site. Wait until the problem clears or contact the remote site system administrator.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8565I **Unimplemented TCP/IP request**

Explanation

TCPIP has received a request for a function that has not been implemented.

System action

TCPIP continues.

Operator response

If the specified request should be implemented in TCPIP, contact the system programmer or administrator.

System programmer response

If the request should be updated in your *hlq*.PROFILE.TCPIP data set, open the data set, set up the specified request in the proper area of the data set and reinitialize TCPIP. For more information about the *hlq*.PROFILE.TCPIP data set see z/OS Communications Server: IP Configuration Reference.

Module

various.

Procedure name

various.

EZA8566I **Destination host is not known**

Explanation

The host address that you are attempting to reach is not known.

System action

TCPIP continues. The program halts or the connection is not established.

Operator response

Check the address and make sure you used the proper syntax and that the address exists for the remote host that you are attempting to reach. If the problem persists, contact the remote host administrator.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8567I **Destination network is unreachable**

Explanation

The network address that you entered is an unreachable network address.

System action

TCPIP continues. The program or the connection abends.

Operator response

Check that the IP addresses are correct in the *hlq*.PROFILE.TCPIP data set. Use the BEGINROUTES statement to add the destination to the configuration data set if necessary. See the z/OS Communications Server: IP Configuration Reference for more information about the BEGINROUTES statement.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8568I **Unspecified connection**

Explanation

In the case of SMTP and FTS server, this could indicate an error with the code itself. The connection is not established.

System action

TCPIP continues.

Operator response

If the error specified is from any application other than the FTP server or SMTP, wait for data or ask the remote site to send you an acknowledgment packet to test your connection. If the error is with the SMTP or the FTP server code, refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the application code. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application program.

Module

various.

Procedure name

various.

EZA8569I **Client address space has too little storage**

Explanation

The virtual storage of a client address is too small for TCPIP to continue processing.

System action

TCPIP halts.

Operator response

Wait until TCPIP resumes. If TCPIP does not resume, contact the system programmer or the IBM Software Support Center.

System programmer response

Increase the virtual storage for client.

Module

various.

Procedure name

various.

EZA8570I **Foreign host disagreed on security or precedence**

Explanation

The other host has rejected the connection because of discrepancies in either the security or precedence of the open attempt.

System action

TCPIP halts. The calling program halts.

Operator response

This error should never occur. If it does it indicates an error with the server FTP code itself. Refer this error to the system programmer or the IBM Software Support Center.

System programmer response

Parse the code with trace on for the specific application where the error occurred. Look for any indications of possible errors and correct them. Recompile after any corrections are made and test the application through TCPIP.

Module

various.

Procedure name

various.

EZA8571I **No local X.25 virtual circuits available**

Explanation

There are no available X.25 virtual circuits for access. This error results from problems at the remote site.

System action

TCPIP continues. The program halts or the connection is not established.

Operator response

Wait until the problem clears, or call the remote site administrator for assistance.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8575I **Client has ended TCP/IP service**

Explanation

The client has ended TCPIP either intentionally or unintentionally.

System action

TCPIP continues. The connection abended.

Operator response

This error occurs from the client site. Wait until they restart TCPIP or call the site administrator for assistance.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8576I **TCP cannot handle any more connections now**

Explanation

TCPIP has no more resources available to handle new connections. This is usually a problem with the remote site not having enough resources available to handle your connection request. In the case of the TCPIP engine, this could indicate the address was not specified or there is not enough room in the TRANSLATE table in your *hlq*.PROFILE.TCPIP data set. A connection is not established.

System action

TCPIP continues.

Operator response

If the problem is with the remote site, wait until a connection becomes available. If the problem is with the engine, either add the address to your TRANSLATE table or make room for the address in your *hlq*.PROFILE.TCPIP data set. For more information about the TRANSLATE table and the *hlq*.PROFILE.TCPIP data set, see z/OS Communications Server: IP Configuration Reference.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8577I **Invalid local address for UDP**

Explanation

The local address specified through UDP is not valid. A connection is not established.

System action

TCPIP continues.

Operator response

Verify that the address exists for UDP. Check your address for any syntax errors and try your address again. If you continue to experience problems, contact the IBM Software Support Center.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8579I **Address unspecified when specification necessary**

Explanation

The connection you attempted to access through UDP requires that you specify a valid address. A connection is not established.

System action

TCPIP continues.

Operator response

Check the syntax and the address. Make sure you specify an address as required by UDP connections and reissue your connection request.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8580I **Port unspecified when specification necessary**

Explanation

The connection you attempted to access through UDP required that you specify a valid port. A connection is not established.

System action

TCPIP continues.

Operator response

Check the syntax of your connection request. Make sure that you specify a port as required for UDP connections and reissue your connection.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8582I **FSend still pending on this connection**

Explanation

There is already an FSEND (file in send status) function pending on the connection you are attempting to use.

System action

TCPIP continues.

Operator response

The problem might be with the remote host. Call the system administrator at the remote host for assistance or wait until the current FSEND clears and reissue your request.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8583I **Message from MVS host to S/1 is too large**

Explanation

The Series/1 processor received a message from the MVS host system that is too large to process. The message is dropped.

System action

TCPIP continues.

Operator response

Resend the message using smaller packets of data or decrease the size of the message.

System programmer response

No action is needed.

Module

TO18S1P

Procedure name

To1822S1RuptHandler

EZA8584I **IO write error on the 370 Channel**

Explanation

TCPIP has encountered an I/O error on a System/370 channel.

System action

TCPIP continues.

Operator response

Check the channel adapter for the 1822 interface and make sure that all connections are attached correctly. Also make sure that the hardware is switched on and is online. If you continue to receive this error, check your hardware maintenance guide for more troubleshooting information.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8585I **Obtained unknown sense data**

Explanation

TCPIP does not recognize the data sent to or from the 1822 Series/1 device.

System action

TCPIP continues.

Operator response

If this error persists, check your connections and make sure that they are properly attached. It might be necessary to shut down the device and restart the device to synchronize the connection. Check your hardware maintenance manual for more assistance.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8591I **Series/1 processor is unavailable**

Explanation

TCPIP is getting device errors from the Series/1 channel because the Series/1 processor is either shut down or offline. The connection to the Series/1 is abended.

System action

TCPIP continues.

Operator response

Make sure all the connections to the Series/1 are properly attached. Shut down and restart the device if necessary.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8592I **Interface Message Processor is unavailable**

Explanation

The 1822 Interface Message Processor (IMP) is not ready or offline. The IMP is down.

System action

TCPIP continues.

Operator response

Check your hardware connections and make sure that they are properly attached. Also make sure that the device is online. Restart the device if necessary. If the problem persists, consult your hardware manual for more troubleshooting information.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8593I **Foreign host is dead or unreachable**

Explanation

The other host cannot be reached by TCPIP either because the other host is shut down or the first hop address or network address is not valid.

System action

TCPIP continues.

Operator response

Check the BEGINROUTES statement in your *hlq*.PROFILE.TCPIP data set and make sure that the first hop and network addresses for the other host are defined correctly. For more information about the BEGINROUTES statement see z/OS Communications Server: IP Configuration Reference.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8594I **More than 8 outstanding messages to that host**

Explanation

The Information Message Processor (IMP) has detected more than 8 messages for that host. This could indicate that the host is not receiving because it is shut down or experiencing problems.

System action

TCPIP continues.

Operator response

Try pinging the host to see if it is online. If it is, then it might be that it is slow in receiving its envelopes. Call the host's system administrator for help if necessary.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8595I **Initiating restart, Series/1 thinks host is down**

Explanation

The Series/1 processor is restarting the host because it has not received acknowledgments from the host.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8596I **Series/1 application program is not available**

Explanation

The 1822 Series/1 application is not ready or available. A connection is not established.

System action

TCPIP continues.

Operator response

Check the connections to the Series/1 channel and make sure that they are properly attached. Shut down the driver and start it again. If the system does not respond, contact the IBM Software Support Center.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8597I **Series/1 ready to receive datagram**

Explanation

The 1822 Series/1 processor is now up and ready to receive data. The connection is established.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8598I **X25 Driver in loopback mode**

Explanation

The X.25 interface driver is in a loopback mode. The X.25 interface cannot transmit or receive data.

System action

TCPIP continues.

Operator response

Check the X.25 driver modem hardware and reset it to DATA if it is LOOPBACK. If the hardware is OK, stop and restart the driver through TCPIP. If the problem persists, contact the IBM Software Support Center.

System programmer response

Assist the user as required.

Module

various.

Procedure name

various.

EZA8599I **Connection dropped by operator**

Explanation

The operator intentionally dropped the connection to TCPIP.

System action

TCPIP halts.

Operator response

Reinitiate TCPIP if necessary.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8600I **Error opening or reading data set**

Explanation

TCPIP cannot open or read a Translate table data set. Translate tables are used to convert transmitted data from EBCDIC to ASCII or ASCII to EBCDIC.

System action

TCPIP continues. The calling program continues.

Operator response

Make sure that the translate table data set is available to TCPIP. If available, make sure that the data set does not contain corrupted information. For more information about translate tables see z/OS Communications Server: IP Configuration Reference.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8601I **Data set format invalid**

Explanation

The translate table data set has been recognized by TCPIP as a having a premature end of file (EOF). This indicates that the data set is a null file or that the data set has corrupted data. The translate table data set is not read.

System action

TCPIP continues.

Operator response

Check the translate table data set and make sure that the data is in the correct format and that the data set is not blank. For more information about translate tables see z/OS Communications Server: IP Configuration Reference.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8602I **Offload box error**

Explanation

TCPIP has received an error from the Offload device that TCPIP does not recognize.

System action

TCPIP halts.

Operator response

Restart the Offload device and TCPIP. If the problem persists, contact the IBM Software Support Center.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8603I **Offload box restart**

Explanation

The Offload device has been restarted.

System action

TCPIP halts.

Operator response

Wait until the Offload device has been completely initiated before using TCPIP.

System programmer response

No action is needed.

Module

various.

Procedure name

various.

EZA8604I **Offload box down**

Explanation

The Offload device is not responding to TCPIP.

System action

TCPIP halts.

Operator response

Restart the Offload device and wait until it initiates TCPIP before attempting to use it. If this problem persists contact the IBM Software Support Center.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

EZA8624E ***** Unknown TCP return code = *CallReturn* *****

Explanation

TCPIP has encountered a return code that it does not recognize.

System action

TCPIP continues.

Operator response

Call the IBM software Support Center for assistance.

System programmer response

Assist the operator as required.

Module

various.

Procedure name

various.

Note: The following messages display the call type returned from the procedure SayCliCa, which is called to determine the type of call issued by a client.

EZA8625I **Abort TCP**

Explanation

A client is using the procedure TcpAbort to close TCP service.

System action

TCP service to the client ends. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpAbort

EZA8626I **Begin TCP/IP service**

Explanation

A client is using the procedure StartTcpIpService to initiate TCP/IP service.

System action

TCP/IP service is initiated for the client.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

StartTcpIpService

EZA8627I **Close TCP**

Explanation

A client has issued a request to close TCP service.

System action

TCP service to the client ends. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8628I **Close UDP**

Explanation

A client has issued a request to close UDP service.

System action

UDP service to the client ends. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8629I **End TCP/IP service**

Explanation

A client has issued a request to end TCP/IP service.

System action

TCP/IP service to the client ends. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8630I **Handle notice**

Explanation

The procedure ClientHandle has been called to handle an event notification. ClientHandle checks a client activity and initiates the proper procedure to handle the notification.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCPREQU

Procedure name

TcpRequest

EZA8631I **Is host local?**

Explanation

A client has requested verification of a local address by the procedure ClientIsHostLocal to determine whether a host address is local. Address verification is performed by the procedure ClientIsHostLocal.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCPREQU

Procedure name

TcpRequest

EZA8632I **Monitor command**

Explanation

A client has called the procedure MonCommand, which instructs the TCPIP address space's monitor process to open a data set and execute the commands found there. This is the principal means of updating network and address tables while the TCPIP address space is in use. MonCommand opens the data set and executes the commands found there.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

MonCommand

EZA8633I **Monitor query**

Explanation

A client has called the procedure MonQuery, which is used to check the status of the TCPIP address space's monitor process.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

MonQuery

EZA8634I **No client call**

Explanation

A virtual machine communication facility (VMCF) packet header did not contain a client call.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMVERT

Procedure name

CallToRequest

EZA8635I **Open TCP**

Explanation

A client has issued a call to the procedure ClientOpen, which establishes connections to TCP for clients, to establish TCP service. TCP service is established for the client.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCPREQU

Procedure name

TcpRequest

EZA8636I **Open UDP**

Explanation

A client has issued a call to the procedure UdpOpen, which establishes UDP connections, to open UDP service. UDP service to the client is opened.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

UdpOpen

EZA8637I **Option TCP**

Explanation

A client has issued a call to the procedure ClientOption, which sets client TCP options, to specify options for a TCP connection. ClientOption completes execution and returns.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCPREQU

Procedure name

TcpRequest

EZA8638I **Receive TCP**

Explanation

A client is using the procedure TcpReceive to notify the host that it is ready to receive TCP data.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpReceive

EZA8639I **Receive UDP**

Explanation

A client is using the procedure UdpReceive to notify the host that it is ready to receive UDP datagrams.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

UdpReceive

EZA8640I **NReceive UDP**

Explanation

A client is using the procedure UdpNReceive to notify the host that it is ready to receive UDP datagrams. UdpNReceive allows the client to specify the maximum size for the datagrams it will receive. This parameter is not supported under UdpReceive.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

UdpNReceive

EZA8641I **Status UDP**

Explanation

A client has called the procedure UdpStatus to check the status of a UDP connection.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

UdpStatus

EZA8642I **Option UDP**

Explanation

A client has issued a call to the procedure UdpOption, which sets client UDP options, to specify options for a UDP connection.

System action

UdpOption completes execution and returns. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

UdpOption

EZA8643I **Send TCP**

Explanation

A client is using the procedure TcpSend to send data normally over a TCP connection.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpSend

EZA8644I **Send UDP**

Explanation

A client is using the procedure UdpSend to send data grams normally over a UDP connection.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

UdpSend

EZA8645I **Status TCP**

Explanation

A client is using the procedure TcpStatus to check the status of a connection to TCP.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpStatus

EZA8646I **FReceive TCP**

Explanation

A client is using the procedure TcpFReceive to receive data pushed over a TCP connection.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpFReceive

EZA8647I **FSend TCP**

Explanation

A client is using the procedure TcpFSend to send data over a TCP connection before the output buffer is full. This function is not supported under TcpSend, which is used to send data normally over TCP connections.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpFSend

EZA8648I **Close RawIP**

Explanation

A client is closing a raw IP connection. All data associated with the connection, including any data queued to that connection will be destroyed.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

SmfSend, RawIpClose

EZA8649I **Open RawIP**

Explanation

A client is using the procedure RawIpOpen to open a raw IP connection to the host.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

RawIpOpen

EZA8650I **Receive RawIP**

Explanation

A client is using the procedure RawIpReceive to tell the network that it is ready to receive raw IP packets.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

RawIpReceive

EZA8651I **Send RawIP**

Explanation

A client is using the procedure RawIpSend to send raw IP packets over the network normally.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

RawIpSend

EZA8652I **Ping request**

Explanation

A client is using the procedure PingRequest to perform a PING to test continuity across the network. For more information on the PING command, see the z/OS Communications Server: IP User's Guide and Commands.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

PingRequest

EZA8669E ***** Invalid call: ord = *call_number* *****

Explanation

A client has issued a call that TCPIP does not recognize. This indicates a programming error. The call is refused.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the client to determine why it is sending incorrect calls.

Module

CMPRCOM

Procedure name

SayCliCa

Note: The following messages display the connection state returned by the function SayConSt, which is called to determine the state of a connection.

EZA8670I **Connection closing**

Explanation

The connection for which SayConSt was called is closing.

System action

TCPIP continues.

Operator response

Restart the connection if necessary.

System programmer response

Assist the user as necessary.

Module

varies

Procedure name

varies

EZA8671I **Listening**

Explanation

The connection for which SayConSt was called is currently in a listening state.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8672I **Nonexistent**

Explanation

The connection for which SayConSt was called does not exist.

System action

TCPIP continues.

Operator response

Start the connection if necessary.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8673I **Open**

Explanation

The connection for which SayConSt was called is in an Open state.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8674I **Receiving only**

Explanation

The connection for which SayConSt was called is in a Receiving Only state.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8675I **Sending only**

Explanation

The connection for which SayConSt was called is in a Sending Only state.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8676I **Trying to open**

Explanation

TCPIP is attempting to open the connection for which SayConSt was called.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8679E ***** Invalid state: ord = *state* *****

Explanation

SayConSt encountered a state it did not recognize. The state is displayed in the message.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the devices on each end of the connection to determine why they are sending incorrect state information. For more information about valid connection states for TCPIP, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Module

varies

Procedure name

varies

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

EZA8680I **Buffer space available**

Explanation

TCPIP has received notification that buffer space is available to send or receive data.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8681I **Connection state changed**

Explanation

TCPIP has received notification that the state of a connection has changed.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8682I **Data delivered**

Explanation

TCPIP has received notification that data has been delivered.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8683I **User-defined notification**

Explanation

TCPIP has received a user-defined notification.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8684I **Datagram space available**

Explanation

TCPIP has received notification that space is available to send or receive data grams.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8685I **Urgent pending**

Explanation

TCPIP has received notification that an urgent transmission is about to be sent.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8686I **UDP data delivered**

Explanation

TCPIP has received notification that UDP data has been delivered.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8687I **UDP datagram space available**

Explanation

TCPIP has received notification that space is available to send or receive UDP datagrams.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8688I **Other external interrupt received**

Explanation

TCPIP has received an external interrupt from the operating system. The appropriate interrupt handler is called to deal with the external interrupt.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8689I **User delivers line**

Explanation

A user has sent data to TCPIP. The data is received and processed.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8690I **User wants attention**

Explanation

A user has requested service from TCPIP. TCPIP delivers the requested service to the user.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8691I **Timer expired**

Explanation

A timer has expired on a process or event.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

Processing resumes. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8692I **FSend response**

Explanation

A client has sent notification that it is using the procedure TcpFSend to push data over a TCP connection. The data is sent.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8693I **FReceive error**

Explanation

A client was unable to receive data because the connection to which the data was sent was not in the correct state to receive it. TCPIP will handle the resulting interrupt and retransmit the data.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8694I **RawIP packets delivered**

Explanation

Raw IP packets have been delivered to a client.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8695I **RawIp packet space available**

Explanation

TCPIP has received notification that buffer space is available to receive raw IP packets.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8696I **IUCV interrupt**

Explanation

TCPIP has received an interrupt from the inter&hyphen user communication vehicle (IUCV). The IUCV interrupt handler is called to deal with the interrupt.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8697I **I/O interrupt**

Explanation

TCPIP has received an input/output (I/O) interrupt. The I/O interrupt handler is called to deal with the interrupt.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8698I **Resources available for TcpOpen**

Explanation

The necessary resources are currently available for the procedure TcpOpen, which opens a connection to TCP.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8699I **Resources available for UdpOpen**

Explanation

The necessary resources are available for the procedure UdpOpen, which opens a connection to UDP.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

A UDP connection is opened. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8700I **Ping response or timeout**

Explanation

TCPIP has received a response from a PING request, used to check connectivity across the network. If the response was not a timeout, it indicates that the host to which the PING was directed is reachable over the network.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8701I **SMASH received**

Explanation

A short message has been received and queued to the appropriate connection.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8706E ***** Invalid notification: ord = *notification* *****

Explanation

SayNotEn has received a notification that it does not recognize. The notification is ignored.

Note: This message displays notifications returned by the function SayNotEn, which is used to display event notifications sent by the TCPIP address space.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the device that sent the notification to determine why it is sending incorrect notifications.

Module

varies

Procedure name

varies

EZA8707I **Routine**

Explanation

TCPIP has received a packet with a precedence value of routine.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8708I **Priority**

Explanation

TCPIP has received a packet with a precedence value of priority.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8709I **Immediate**

Explanation

TCPIP has received a packet with a precedence value of immediate.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8710I **Flash**

Explanation

TCPIP has received a packet with a precedence value of flash.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8711I **Flash override**

Explanation

TCPIP has received a packet with a precedence value of flash override.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8712I **Critic/ECP**

Explanation

TCPIP has received a packet with a precedence value of Critic/ECP.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8713I **Internetwork control**

Explanation

TCPIP has received an internetwork control packet.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8714I **Network control**

Explanation

TCPIP has received a network control packet.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8717E ***** Invalid precedence: ord = *precedence* *****

Explanation

TCPIP has received a packet with a precedence value that it does not recognize. The packet is discarded.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the sending device to determine why it is sending packets with an incorrect precedence value.

Module

varies

Procedure name

varies

Note: The following messages display the protocol type returned by the function SayProTy, which is called to determine the protocol type of an incoming packet.

EZA8718I **ICMP**

Explanation

TCPIP has received a packet with a protocol type of internet control message protocol (ICMP).

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8724I **TCP**

Explanation

TCPIP has received a transmission control protocol (TCP) packet.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8734I **UDP**

Explanation

TCPIP has received a user datagram protocol (UDP) packet.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8736I ***protocol***

Explanation

TCPIP has received a message with the indicated protocol.

Note: This message displays the precedence values returned by the function SayPreTy, which is used to display the precedence value of an incoming packet. For more information about precedence values, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8737I **Specific authorization**

Explanation

A packet has been received with a specific authorization.

Note: This message displays the authority returned by the function SayVmcAu, which is called to determine the authority level of a packet transmitted by the virtual machine communication facility (VMCF)

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8738I **Priority**

Explanation

A packet has been received with a priority authorization.

Note: This message displays the authority returned by the function SayVmcAu, which is called to determine the authority level of a packet transmitted by the virtual machine communication facility (VMCF)

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8739I **Allow short message facility**

Explanation

A packet has been received authorizing use of the short message facility of the virtual machine communication facility (VMCF).

Note: This message displays the authority returned by the function SayVmcAu, which is called to determine the authority level of a packet transmitted by the virtual machine communication facility (VMCF)

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8745E ***** Invalid authorization: *authority* *****

Explanation

The virtual machine communication facility (VMCF) has received a packet with an authority that it does not recognize.

Note: This message displays the authority returned by the function SayVmcAu, which is called to determine the authority level of a packet transmitted by the virtual machine communication facility (VMCF)

System action

The packet is ignored. TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the device that sent the packet to determine why it is sending packets with incorrect authorities.

Module

varies

Procedure name

varies

EZA8746I **Authorize**

Explanation

The virtual machine communication facility has received an authorize call from a client.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8747I **Unauthorize**

Explanation

The virtual machine communication facility (VMCF) has received an unauthorize function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8748I **Send**

Explanation

The virtual machine communication facility (VMCF) has received a send function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8749I **Send/Receive**

Explanation

The virtual machine communication facility (VMCF) has received a send/receive function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8750I **SendX**

Explanation

The virtual machine communication facility (VMCF) has received a sendx function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8751I **Receive**

Explanation

The virtual machine communication facility (VMCF) has received a receive function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8752I **Cancel**

Explanation

The virtual machine communication facility (VMCF) has received a cancel function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8753I **Reply**

Explanation

The virtual machine communication facility (VMCF) has received a reply function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8754I **Quiesce**

Explanation

The virtual machine communication facility (VMCF) has received a quiesce function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8755I **Resume**

Explanation

The virtual machine communication facility (VMCF) has received a resume function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8756I **Identify**

Explanation

The virtual machine communication facility (VMCF) has received an identify function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8757I **Reject**

Explanation

The virtual machine communication facility (VMCF) has received a reject function call from TCPIP.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8758E ***** Invalid VMCF function: *function* *****

Explanation

The virtual machine communication facility (VMCF) has received a call for a function it does not recognize. The call is ignored.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the sending device to determine why it is making incorrect VMCF function calls.

Module

varies

Procedure name

varies

EZA8915I **version: *version***

Explanation

This message displays the internet addressing protocol being used.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8916I **Internet Header Length: *length* = *number* bytes**

Explanation

This message displays the internet header length being used for IP packets and the equivalent length in bytes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8917I **Type of Service: Precedence = *precedence*, *performance_emphasis*, *** garbage in reserved bits = *incorrect_data***

Explanation

These messages indicate the type of service being delivered by an IP datagram, the precedence value, and the performance emphasis. The performance emphasis can be either low delay, high throughput, or high reliability. The message “garbage in reserved bits” is displayed only if incorrect data is encountered in a reserved field of the IP header. The data will be ignored.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8924I **Total Length: *number* bytes**

Explanation

This message displays the total length of the IP datagram.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8925I **Identification: *identification* Flags: *don't fragment/may fragment*, *more fragments/last fragment* fragment offset: *length* = *number* bytes**

Explanation

This message displays the identification of an IP datagram and fragmentation information for that datagram.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8933I **Time To Live:** ***time*** **Protocol:** ***protocol*** **Header Checksum:** ***checksum*** **Source Address:** ***address*** **Destination Address:** ***address***

Explanation

This message gives the time to live, protocol type, header checksum, and source and destination addresses for an IP datagram.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8936I **Protocol:** ***protocol***

Explanation

This message indicates the protocol being used by TCPIP.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8937I **Header CheckSum: *checksum***

Explanation

This message displays the header checksum from a packet. The header checksum is used to check packet integrity.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8938I **Source Address: *address***

Explanation

This message displays the address from which a packet was sent.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8939I **Destination Address: *address***

Explanation

This message displays the address to which a packet was sent.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8961I **net address = *address*, port = *port_type***

Explanation

This message displays the internet address and port type of a port.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8966I **Connection: *connection***

Explanation

This message displays the current connection.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8967I **Open Attempt Timeout: *time***

Explanation

The current connection will wait for the indicated length of time for the other side of the connection to respond to an open request.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8968I **Wait Forever**

Explanation

The value of the OpenAttemptTimeout is WAITForever, indicating that the host will wait for a connection until the connection is opened.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8969I ***openattempttimeout***

Explanation

This message displays the value of the OpenAttemptTimeout, which indicates the length of time, in seconds, that the host will wait for a connection to be opened before canceling the attempt.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8970I **Security: *security***

Explanation

This message gives the security status of a connection.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8971I **Compartment: *compartment***

Explanation

This message indicates the compartment to which a connection is assigned.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8972I **Precedence: *precedence***

Explanation

This message displays the precedence type of an IP datagram.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8973I **Bytes To Read: *number***

Explanation

This message displays the number of bytes to be read in an IP datagram.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8974I **Unacked Bytes: *number***

Explanation

This message displays the number of unacknowledged bytes in an IP datagram.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8975I **Connection State: *connection_state* Local Socket: *socket* Foreign Socket: *socket***

Explanation

This message displays the connection state for the current connection and the local and foreign sockets being used for the connection. For more information about the connection state, see the message EZA*nnnn*, where *nnnn* is the 4-digit number in parentheses displayed in this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8980I **V1: *vmcf_response_code***

Explanation

This message displays the response code taken from the packet header of an external virtual machine communication facility (VMCF) packet. The response code indicates the status of the external end of the VMCF connection. For more information about VMCF response codes, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8981I **V1: *VCMPFLG1***

Explanation

This message displays the VCMPFLG1, which is used to specify options associated with a particular function. The flag byte can be set to the following values:

Flag Value
Indicates

VMCPAUTS (X‘80')
An AUTHORIZE SPECIFIC request.

VMCPPRTY (X‘40')
For SEND, SEND/RECV, SENDX, and IDENTIFY requests, a PRIORITY message request. For an AUTHORIZE request, it indicates an AUTHORIZE PRIORITY request. PRIORITY messages cannot be sent to another machine unless it has been authorized to receive them.

VMCPSMSG (X‘20)
The virtual machine accepts messages sent via the SMSG command.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8982I **V2: *VCMPFLG2***

Explanation

This message displays the hexadecimal value indicating the virtual machine communication facility (VMCF) function request taken from the VMCF packet header. The flag is reserved, and therefore should be X‘00' initially.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8983I **Function: *VMCF_function***

Explanation

This message displays the virtual machine communication facility (VMCF) function taken from the VMCF packet header. For more information about VMCF functions, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8984I **Message ID: *message_identifier***

Explanation

This message displays the message identifier taken from the virtual machine communication facility (VMCF) packet header. For more information about VMCF message identifiers, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8985I **Login name: *name***

Explanation

This message displays the login name associated with a virtual machine communication facility (VMCF) packet. For more information about VMCF, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8986I **Vaddress A: *address***

Explanation

This message displays the VMCPVADA register taken from the virtual machine communication facility (VMCF) packet header. This register contains one of four addresses, depending on which VMCF function is requested.

For SEND, SEND/RECV, and SENDX requests, VMCPVADA contains the address of the source address space data. For RECEIVE requests, VMCPVADA contains the address of a sink address space RECEIVE buffer. For REPLY requests, VMCPVADA contains the address in sink address space where REPLY data is located. For an AUTHORIZE request, VMCPVADA contains the address of an external interrupt buffer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8987I **Length A: *length***

Explanation

This message specifies the length of the type of data specified by *Vaddress A*.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8988I **Vaddress B: *VMCPVADB***

Explanation

This message displays the contents of the VMCPVADB register taken from the virtual machine communication facility (VMCF) packet header. This register contains the address of a source address space's REPLY buffer. The length of the buffer is specified by *VMCPLENB*.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8989I **Length B: *VMCPLENB***

Explanation

This message displays the VMCPLENB register taken from the virtual machine communication facility (VMCF) packet header. This register indicates the length of the REPLY buffer specified in *VMCPVADB*.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8990I **AnInteger: *integer_value***

Explanation

The meaning of the value displayed in AnInteger depends on the type of virtual machine communication facility (VMCF) function call contained in the packet:

For the function:
AnInteger represents:

PINGreq
The value of the timeout being used for the PING request.

PINGresponse
The return code from the PING procedure.

UDPdatagramDELIVERED
The length of the entire datagram excluding the UDP header.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8992I **Connection: Unspecified**

Explanation

This message indicates that connection by which a virtual machine communication facility (VMCF) packet was sent is unspecified.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8993I **Connection: *connection***

Explanation

This message indicates the connection by which a virtual machine communication facility (VMCF) packet was sent.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8994I **Call: *call_code***

Explanation

This message displays the call code from a virtual machine communication facility (VMCF) packet header.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA8995I **Return code: *SayCalRe(return_code)***

Explanation

This message displays the return code from a virtual machine communication facility (VMCF) packet header.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Module

varies

Procedure name

varies

Chapter 10. EZA9*xxxx* messages

EZA9000I **V1:** ***vmcf_response_code***

Explanation

This message displays the response code taken from the packet header of an external virtual machine communication facility (VMCF) packet. The response code indicates the status of the external end of the VMCF connection. For more information about VMCF response codes, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9001I **V1:** ***VCMPFLG1***

Explanation

This message displays the VCMPFLG1, which is used to specify options associated with a particular function. The flag byte can be set to the following values:

VMCPAUTS (X‘80’)
An AUTHORIZE SPECIFIC request.

VMCPPRTY (X‘40’)
For SEND, SEND/RECV, SENDX, and IDENTIFY requests, a PRIORITY message request. For an AUTHORIZE request, it indicates an AUTHORIZE PRIORITY request. PRIORITY messages cannot be sent to another machine unless it has been authorized to receive them.

VMCPSMSG (X‘20)
The virtual machine accepts messages sent via the SMSG command.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9002I **V2: *VCMPFLG2***

Explanation

This message displays the hexadecimal value indicating the virtual machine communication facility (VMCF) function request taken from the VMCF packet header. The flag is reserved, and therefore should be X'00' initially.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9003I **Function: *VMCF_function***

Explanation

This message displays the virtual machine communication facility (VMCF) function taken from the VMCF packet header. For more information about VMCF functions, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9004I **Message ID: *message_identifier***

Explanation

This message displays the message identifier taken from the virtual machine communication facility (VMCF) packet header. For more information about VMCF message identifiers, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9005I **Login name: *name***

Explanation

This message displays the login name associated with a virtual machine communication facility (VMCF) packet. For more information about VMCF, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9006I **Vaddress A: *address***

Explanation

This message displays the VMCPVADA register taken from the virtual machine communication facility (VMCF) packet header. This register contains one of four addresses, depending on which VMCF function is requested.

For SEND, SEND/RECV, and SENDX requests, VMCPVADA contains the address of the source address space data. For RECEIVE requests, VMCPVADA contains the address of a sink address space RECEIVE buffer. For REPLY requests, VMCPVADA contains the address in sink address space where REPLY data is located. For an AUTHORIZE request, VMCPVADA contains the address of an external interrupt buffer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9007I **Length A: *length***

Explanation

This message specifies the length of the type of data specified by *Vaddress A*.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9008I **Vaddress B: *VMCPVADB***

Explanation

This message displays the contents of the VMCPVADB register taken from the virtual machine communication facility (VMCF) packet header. This register contains the address of a source address space's REPLY buffer. The length of the buffer is specified by *VMCPLENB*.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9009I **Length B: *VMCPLENB***

Explanation

This message displays the VMCPLENB register taken from the virtual machine communication facility (VMCF) packet header. This register indicates the length of the REPLY buffer specified in *VMCPVADB*.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9010I **AnInteger: *AnInteger***

Explanation

The meaning of the value displayed in AnInteger depends on the type of virtual machine communication facility (VMCF) function call contained in the packet:

PINGreq
The value of the timeout being used for the PING request.

PINGresponse
The return code from the PING procedure.

UDPdatagramDELIVERED
The length of the entire datagram excluding the UDP header.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9011I **Connection: Unspecified**

Explanation

This message indicates that connection by which a virtual machine communication facility (VMCF) packet was sent is unspecified.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9012I **Connection: *connection***

Explanation

This message indicates the connection by which a virtual machine communication facility (VMCF) packet was sent.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9013I **Call: *call_code***

Explanation

This message displays the call code from a virtual machine communication facility (VMCF) packet header.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9014I **Return code: *SayCalRe(return_code)***

Explanation

This message displays the return code from a virtual machine communication facility (VMCF) packet header.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Module

varies

Procedure name

varies

EZA9020I **VMCF ret code is *return_code*. VMCF header follows:**

Explanation

This message displays the return code from a virtual machine communication facility (VMCF) packet. It precedes the actual VMCF header.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Use the return code displayed in the message and the list of VMCF return codes in the z/OS Communications Server: IP Diagnosis Guide to determine whether an error has occurred and respond as indicated by the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

Module

varies

Procedure name

varies

EZA9025I **VMCF interrupt header:**

Explanation

This message precedes the printing of the virtual machine communication facility (VMCF) interrupt header.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The VMCF interrupt header is printed. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9030I ***source_address* -> *destination_address* V*version* IHL=*header_length* Len=*packet_length* ID=*identification* Frag Offset: *fragmentation_offset=number* bytes (More) Protocol=*SayProTy(protocol)***

Explanation

This message displays information about an incoming IP datagram. IHL is the internet header length, Len is the total length of the datagram in bytes. Frag Offset is the fragmentation offset, and (More) indicates that there are more fragments of this datagram to come.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

Note: Messages EZA9042I-EZA9050I display the value returned by the function QuickNotificationSetPrint, which returns a 2 letter code indicating the type of notification that has been received.

EZA9042I **Bs**

Explanation

A notification has been received indicating that buffer space is available.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9043I **Cs**

Explanation

A notification has been received indicating that the state of a connection has changed.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9044I **Dd**

Explanation

A notification has been received indicating that data has been delivered.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9045I **Un**

Explanation

A user-defined notification has been received.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9046I **Ds**

Explanation

A notification has been received indicating that datagram space is available.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9047I **Up**

Explanation

A notification has been received indicating that an urgent message is pending.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9048I **Ud**

Explanation

Notification of a user delivered line has been received.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9049I **Uw**

Explanation

Notification that a user wants attention has been delivered.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9050I *****

Explanation

An undefined notification has been received.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Check the sending device to determine why it is sending undefined notifications.

Module

varies

Procedure name

varies

EZA9060I ***address.SayPorTy(port)***

Explanation

This message displays the address and port type of a socket.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9070I ***#connection SayConSt(state)* Open timeout=*time* To read=*bytes_to_read* Unacked=*unacknowledged_bytes* Local= Foreign=**

Explanation

This message indicates the status of a connection, including the connection state, open timeout, number of bytes to read, number of unacknowledged bytes, and the address and port type of the local and remote sockets.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9080I ***SayVmcFu(function)->directory_name* V1=*VMCPFLG1* ID=*message_id* VadA=*VMCPVADA* LenA=*length_of_VMCPVADA* VadB=*VMCPVADB* LenB=*length_of_VMCPVADB* Conn=*connection* Call=*call_code* Return=*SayCalRe(return_code)***

Explanation

This message displays the contents of a virtual machine communication facility (VMCF) packet. V1 is a flag byte used to specify options associated with a particular function. The flag byte can be set to the following values:

VMCPAUTS(X‘80’)
An AUTHORIZE SPECIFIC request.

VMCPPRTY(X‘40’)
For SEND, SEND/RECV, SENDX, and IDENTIFY requests, a PRIORITY message request. For an AUTHORIZE request, it indicates an AUTHORIZE PRIORITY request. PRIORITY messages cannot be sent to another address space unless that address space has been authorized for PRIORITY messages. The SEND and

RESPONSE external interrupts for a PRIORITY message are queued ahead of pending nonpriority external interrupts.

VMCPSMSG(X'20')
The virtual machine accepts messages sent via the SMSG command.

The ID value is the message identifier.

VadA contains one of four addresses, depending upon which VMCF function is requested.

For SEND, SEND/RECV, and SENDX requests, VMCPVADA contains the address of the source address space data. For RECEIVE requests, VMCPVADA contains the address of a sink address space RECEIVE buffer. For REPLY requests, VMCPVADA contains the address in sink address space where REPLY data is located. For an AUTHORIZE request, VMCPVADA specifies the address of the interrupt buffer.

LenA contains the length of data sent by a user, the length of a RECEIVE buffer, or the length of an external interrupt buffer, whichever is specified in the field VMCPVADA.

VMCPVADB contains the address of a source address space's REPLY buffer for a SEND/RECV request. When the sink address space issues a REPLY in response to a SEND/RECV from the source address space, the REPLY data is moved in this buffer.

LenB specifies the length of the buffer being used to move the data indicated in the field VMCPVADB.

Conn specifies the connection being used to send and receive VMCF packets.

Call displays the call code associated with the packet.

Return indicates whether the packet was sent and received successfully.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Module

varies

Procedure name

varies

EZA9200I **Implied Argument lengths:** ***lengths***

Explanation

These trace messages display the size in characters of the arguments submitted.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

DISPLAY_ARGUMENT_SIZES

EZA9202I **filename *data set name***

Explanation

This is the data set name of the message file being used by Falprint to hold text for messages that Falprint generates as write to operator (WTO) messages, which are written to the operator's console. This allows messages from TCPIP or MVS to be displayed even if an overlying application is being used. This message is displayed with message EZA9228I.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

prtmsgw

EZA9203E **MSG *message number* not defined**

Explanation

The specified *message number* is not defined in the message data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Insert the message ID and message text in the message data set.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

printmsgw

EZA9204I **findmsg&colon try again**

Explanation

The message specified was not properly linked to the Falprint message repository. Falprint creates a repository of TCPIP messages that are not displayed while an application is being run on TCPIP and converts them to write to operator (WTO) messages, allowing them to be displayed without interference from an overlying application.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program tries again and continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

findmsg

EZA9205I **searching for weak link: *data_set_name msg_file***

Explanation

Falprint is searching for a weak link between the data set from which message text is being taken and the message file, which is used by Falprint to generate write to operator (WTO) messages, which are written to the operator's console if necessary. A weak link indicates that a message needed by Falprint from the current data set is not duplicated in the message file.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

varies

EZA9206I **Found weak link for message file *name***

Explanation

A message in the indicated data set did not correspond to a message in the message file being constructed by Falprint, creating a weak link between the message files. Falprint builds a message file containing message text from TCPIP and MVS messages that will not be displayed if an application is being run over TCPIP and converts them to write to operator (WTO) messages, allowing them to be displayed.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

Notify the system programmer of this message.

System programmer response

Check the message file and the source data set to determine where the weak link occurred and add the necessary message to the message file.

Module

FAL@PRTC

Procedure name

isweakmsg

EZA9207I **findmsg: looking for message number *message number***

Explanation

The findmsg function, which finds a specific message in a message file, is searching the file for the specified message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

findmsg

EZA9208I **Reading message file: *message file***

Explanation

Falprint is in the process of reading the specified message file. Falprint creates a message file to hold the message identifiers and message text from messages from TCPIP and MVS that will not be displayed due to an application that is being run and converts the messages to write to operator (WTO) messages, which are written to the operator's console, allowing them to be displayed.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9209I **initmsgs read message *message_text message_delimiter***

Explanation

The indicated message text did not contain a delimiter indicating the end of the text. Falprint appends the message delimiter “-n’ to the text to indicate the end of the text.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9210I **Opened dataset *data set***

Explanation

Falprint is using the indicated data set to create a message file containing the message identifiers and message texts for messages from TCPIP and MVS that will not be displayed by normal means due to interference from an application currently running. Falprint converts these messages to write to operator (WTO) messages, allowing them to be displayed while the application is running.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

RES@MVS

Procedure name

mvsfopen

EZA9213I **EZACISSP not found**

Explanation

The program was unable to find the communications interface system (CIS) server.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

Notify the system programmer of this message.

System programmer response

Make sure that the CIS server is started and online. Make sure that the CIS server is reachable through the network.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9214W **extra quote**

Explanation

There is an incorrect number of quotation marks at the end of a line in the message data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Fix the message data set by removing the excess quotation marks.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9215W **Duplicate message file name *name***

Explanation

This message indicates the name specified has already been used for a message data set name.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9218E **Can't open message file or message text name: *data set name***

Explanation

There was a problem opening the message data set or message text name.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Verify the name of the message data set and try again.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9219E **Invalid MARGINS statement in message file.**

Explanation

In the MARGIN statement of the message file, the second margin was smaller than the first margin, or the first margin was assigned a value smaller than 1. This message is also displayed if the MARGINS statement was not in the following form: `MARGINS`(*first margin second margin*)

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

Correct the MARGINS statement in the message file.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9220E **Can't allocate storage for messages**

Explanation

There was an error allocating dynamic storage for the messages.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Increase the region size.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9221E **Missing product in message file: *data set name***

Explanation

The product field is missing from the message data set. The product field contains a 3 letter combination. For example:

- `PRODUCT EZA`
- `PRODUCT EZB`
- `PRODUCT EZY`

For more information on the original message file, ask the system programmer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

See the originally shipped version of the specified data set and verify the name of the product.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9222E **Missing component in message file: *data set name***

Explanation

There was a component field missing from the message data set. The component field is a combination of 3 letters representing individual components. For example:

```
COMPONENT DNS
```

For more information on the message facility, ask the system programmer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

See the originally shipped version of the specified data set and verify the name of the component.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9223E ****** Duplicate Message number ****: *message number***

Explanation

This message indicates that there are 2 or more messages with the same message number in the message data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Delete 1 of the message's text and its corresponding number so there is 1 version of each.

System programmer response

No action is needed.

Module

FAL@PTRC

Procedure name

initmsgs

EZA9224I **Original message: *message text***

Explanation

This message displays the first occurrence of the message with the duplicate message number.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9225I **Duplicate message: *message text***

Explanation

This message displays the second occurrence of the message with the duplicate message number.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9226E **Can't allocate storage for message text**

Explanation

There was an error allocating dynamic storage for the message text.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Increase the size of the region.

Module

FAL@PTRC

Procedure name

initmsgs

EZA9227E **Can't allocate storage for message code**

Explanation

There is not enough dynamic storage for the message code.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Increase the size of the region.

Module

FAL@PRTC

Procedure name

initmsgs

EZA9228I **message number *message number* text *text***

Explanation

This message displays all the messages in the message file and the corresponding message numbers.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

prtmsgs, prtmsgw

EZA9229I **Message *message number* not found**

Explanation

The message number specified was not in the message list, but was called in the source code.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Verify the existence of the message ID and text, and try again.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

fflush

EZA9230E **Processed maximum number of message data sets.**

Explanation

The maximum number of message data sets have been processed. The maximum number of message data sets is 256. Processing is no longer an option.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Do not attempt to process any more message data sets.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

find_msgtxt, update_msgfn

EZA9231I **fal@prt: isdupfn**

Explanation

This message is displayed while debugging is enabled and the isdupfn function is invoked. This function determines if there are duplicate message data sets.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

isdupfn

EZA9232I **fal@prt: isdupfn file already processed**

Explanation

The function that determines if there are duplicate message file names has already been successful.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

isdupfn

EZA9234I **fal@prt: dotimestamp**

Explanation

This message is displayed when debugging is enabled and upon the invocation of the dotimestamp function. This function changes the type of time stamping of all messages.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

dotimestamp

EZA9235I **fal@prt: initprt**

Explanation

This message is displayed when debugging is enabled and upon the invocation of the initprt function. This function determines the status of the message data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initprt

EZA9236I **Message file *data_set_name* rc *return_code***

Explanation

This message displays the name of the processed message data set and the return code.

return_code is the return code from message initialization. If it is zero, then message initialization was successful. If it is a negative number, then a preceding message should have explained the error.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initprt

EZA9237E **EZAMSGL command exceeds *number*-byte buffer: *command***

Explanation

The EZAMSGL command, which is invoked by a user-specified command, exceeded the size of the allocated buffer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

Correct the size of the command in the user-specification. For more information about the EZAMSGL command, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

EZAMSGL

EZA9241I **Using pre-loaded message file *message file name* accessed via EZAMSGP.**

Explanation

Falprint is using the default message library included in the Falprint code FAL@PRTC.C, accessed through the function EZAMSGP, which controls the format of the displayed message. Falprint loads the message file.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

Falprint continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initprt

EZA9242I **Using pre-loaded message file *message file name* accessed via EZAMSGW.**

Explanation

Falprint is using the default message library included in the Falprint code FAL@PRTC.C, accessed through the function EZAMSGW, which controls the format of the displayed message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The program continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FAL@PRTC

Procedure name

initprt

EZA9280E **StartTcpIpService: JOBNAME *name* is not the TCPIP address space.**

Explanation

The specified NETSTAT TCP JOBNAME does not match the JOBNAME of the TCPIP address space.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

NETSTAT ends with error message EZA0216I. TCPIP continues.

Operator response

Reenter NETSTAT with the correct TCPIPJOBNAME as specified in the TCPIP.DATA data set.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

StartTcpIpService

EZA9282I **TcpWaitOpen got unexpected state change: *SayConSt(new_state)***

Explanation

The procedure TcpWaitOpen, which opens a TCP connection for a client when one becomes available, received an unexpected state changed on a connection being monitored. For more information about the new state, see message EZA*nnnn* where *nnnn* is the four-digit number in parenthesis at the end of this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*nnnn* where *nnnn* is the four-digit number in parenthesis at the end of this message.

Module

CMERUPT

Procedure name

TcpWaitOpen

EZA9283I **TcpNameChange called after TcpIp service has begun.**

Explanation

The procedure TcpNameChange, which changes the default directory name to be used by TCPIP, was called after TCPIP service had begun. The default directory name cannot be changed after TCPIP service has begun.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

TcpNameChange

EZA9284I **StartTcpipService: Error *SayCalRe(result)* from GetIdentity**

Explanation

The procedure GetIdentity, which was called to determine the ID of the address space calling the procedure StartTcpipService, was unsuccessful. The reason is indicated by the SayCalRe portion of this message.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Module

CMERUPT

Procedure name

StartTcpIpService

EZA9285I **TIMER TRACE:**

Explanation

This message precedes the printing of a timer trace.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

PrintTimers

EZA9286I **Timer: delta = *timer_change* Datum = *timer_associated_notification***

Explanation

This message displays the elapsed time from an internal timer set by TCPIP. The Datum in this message is the notification associated with the timer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

PrintTimers

EZA9288I **RemoveTimer: Elapsed time is *time***

Explanation

The procedure RemoveTimer has been called to remove a timer that has been running for the indicated length of time.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

RemoveTimer

EZA9290I **Inserting timer: Time = *seconds* Data = *timer_associated_notification***

Explanation

The procedure InsertTimeout is inserting a new timer to replace one that has expired or has been removed. The length of time the timer will run is displayed in the message. The Data displayed in this message is the timer associated notification from the timer that has been removed or has expired.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

InsertTimeout

EZA9292I **SetTimer: Elapsed time is *time***

Explanation

This message is sent to the trace file if the TIMERdebug option was specified. The timer queue is periodically adjusted to reflect the time that has elapsed since the clock comparator was last set. The elapsed time displayed in this message is the time since the clock comparator was last set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

SetTimer

EZA9293I **Setting timeout for *number* Seconds**

Explanation

This message is sent to the trace file if the TIMERdebug option was specified. It indicates that a new timer is being created and set to run for the indicated length of time.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

SetTimer

EZA9294I **InHeader.V2 is *header***

Explanation

This message indicates that the procedure MonQuery, which queries the system monitor, has received an error from the virtual machine communication facility (VMCF) indicating that no buffer space is available to receive the query. The query will be resent when buffer space is available to receive it.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

MonQuery halts. TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMERUPT

Procedure name

MonQuery

EZA9295I **Severe error: Insufficient virtual memory available for TCP-IP client**

Explanation

No virtual storage was available to open a connection to a TCPIP client. The connection is refused and the client will not receive TCPIP service.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The TCPIP address space continues.

Operator response

Notify the system programmer of the error.

System programmer response

Increase the amount of virtual storage available to the host TCPIP address space and restart the connection.

Module

CMERUPT

Procedure name

NoRoom

EZA9340I ***control key***

Explanation

The PA2 or CLEAR key has been pressed. The Full-Screen Package (FSP) returns to CP READ and clears the screen when one of these keys is pressed.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMFSCRN

Procedure name

ReadEnded

EZA9353I **CLIENTconsistency is set for full screen mode.**

Explanation

The CLIENTconsistency has been set for full-screen tracing mode.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMFSCRN

Procedure name

FsIoBegRupt

EZA9354I ***bit number*** **is incorrect in HighTwoBits.**

Explanation

Indicates the bit number in HighTwoBits. This number can be a number in the range 0 - 64.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMFSCRN

Procedure name

FsIoBegRupt

EZA9355I **Terminal Model** ***model number*** **is not supported.**

Explanation

The indicated terminal model number is not supported on this system.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP ends.

Operator response

See the z/OS Communications Server: IP Configuration Reference for more information about technical requirements for TCP/IP for MVS and restart TCPIP.

System programmer response

No action is needed.

Module

CMFSCRN

Procedure name

FsStartUp

EZA9356I **About to begin**

Explanation

This message displays when the debug option is used. The full screen mode has been requested and is about to begin its display.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMFSCRN

Procedure name

FSstart

EZA9403I ***data set name* not found.**

Explanation

The displayed data set name could not be found. When you make changes to your HOSTS.LOCAL data set, you must generate and install new HOSTS.SITEINFO and HOSTS.ADDRINFO data sets.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Use the MAKESITE statement as either a TSO command or a batch job to generate the new data sets. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

CMHOSTN

Procedure name

GetSiteEntry

EZA9404I ***value.value.value.value***

Explanation

Indicates the host's IP address as specified in the NSINTERADDR statement of the TCPIP.DATA data set. This message is displayed with other messages indicating the port number for this IP address. See the z/OS Communications Server: IP Configuration Reference for more information.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMHOSTN

Procedure name

PrintHost

EZA9409E **Syntax or other errors found in *file* file, continuing.**

Explanation

This message is issued when the parser did not find a valid keyword. Syntax or other specification errors were found in the indicated file. The parser state is set to IgnoreState.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

Processing continues.

Operator response

The operator should check that the TCPIP.DATA configuration data set contains the correct syntax and valid information. Correct the input file for correct program operation. See the z/OS Communications Server: IP Configuration Reference for a complete list of the valid keywords.

System programmer response

Assist the user with problem resolution, as required.

Module

CMHOSTN

Procedure name

ReadTCPIPdataFILE

EZA9410E **See *string* line *line number* on or before column *column number***

Explanation

An error has occurred in the indicated line number of the TCPIP.DATA data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Correct the error in the indicated line number and reinitialize TCPIP. See the z/OS Communications Server: IP Configuration Reference for more information.

System programmer response

No action is needed.

Module

CMHOSTN

Procedure name

ReadTCPIPdataFILE

EZA9411E **Unable to allocate/open *dataset* file, continuing.**

Explanation

The application was unable to allocate or unable to open the indicated data set. The reason is indicated by additional messages.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the indicated data set to make sure that it is properly formatted and in storage accessible to the host.

System programmer response

No action is needed.

Module

FTBMVSUB, CMHOSTN

Procedure name

varies

EZA9412I **Unable to allocate/open *data set name* file, continuing.**

Explanation

The attempt to open the HOSTS.SITEINFO data set was unsuccessful. This data set must be generated and installed when changes are made to the HOSTS.LOCAL data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

Use the MAKESITE statement as either a TSO command or a batch job to generate the new data set. See the z/OS Communications Server: IP Configuration Reference for more information.

Module

CMHOSTN

Procedure name

GetSiteEntry

EZA9416E **Failure in procedure getstate for *data_set***

Explanation

The procedure getstate, which is called to determine the status of a connection, was unsuccessful for the indicated data set. This indicates that the data set might not have a connection. The parameter list contained in the indicated data set is not passed to MAKESITE.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the indicated data set to make sure that it is in storage accessible to the host.

Module

CMMAKSI

Procedure name

StartUp

EZA9417E **System error trying to obtain format 1 DSCB for *data_set***

Explanation

The host was unable to determine the format for the data set control block (DSCB) for the indicated data set. CMMAKSI received a return code of 8 from the procedure getstate, which is called to determine the status of a connection. The data set is not read.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Make sure the DSCB is correctly formatted and resubmit the command.

Module

CMMAKSI

Procedure name

StartUp

EZA9418E **Failure using *dataset*, data set is migrated.**

Explanation

The application was unable to use the indicated data set because the data set has been migrated and is inaccessible to the host.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The indicated data set is not accessed. TCPIP continues.

Operator response

Recall the indicated data set. Make sure that it is properly formatted and in storage accessible to the host.

System programmer response

No action is needed.

Module

FTSMVSUB, CMMAKSI, CMHOSTN

Procedure name

varies

EZA9419E **Failure using *data_set*, volume is not mounted.**

Explanation

The MAKESITE command was unable to extract a parameter list from the indicated data set because the volume containing the data set is not mounted.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Mount the correct volume and resubmit the command.

System programmer response

Assist the user as necessary.

Module

CMMAKSI

Procedure name

StartUp.

EZA9420E **Failure using *data_set*, data set is not direct access.**

Explanation

The MAKESITE command was unable to extract a parameter list from the indicated data set because the data set is not a direct access data set. The parameter list is not transferred.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Convert the data set to a direct access data set and resubmit the command.

System programmer response

Assist the user as necessary.

Module

CMMAKSI

Procedure name

StartUp

EZA9421E **Failure using *data_set*, data set is VSAM.**

Explanation

The MAKESITE command was unable to extract a parameter list from the indicated data set because it is a Virtual Storage Access Method (VSAM) data set.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The parameter list is not transferred. TCPIP continues.

Operator response

Make sure that the data set was specified correctly and resubmit the command.

System programmer response

Assist the user as necessary.

Module

CMMAKSI

Procedure name

StartUp

EZA9422E **Failure obtaining *data_set* or invalid data set organization (DSORG)**

Explanation

The MAKESITE command was unable to extract a parameter list from the indicated data set because the data set was specified incorrectly or had an incorrect data set organization. The parameter list is not transferred.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Make sure that the data set was specified correctly and that it is a sequential data set.

System programmer response

Assist the user as necessary.

Module

CMMAKSI

Procedure name

StartUp

EZA9423E **Failure locating *data_set* or invalid record format (RECFM).**

Explanation

The MAKESITE command was unable to extract a parameter list from the indicated data set because the data set was specified incorrectly or had the wrong record format. The MAKESITE command requires a fixed record length. The parameter list is not transferred.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Make sure that the date set was specified correctly and has the proper record format. Correct the record format if necessary.

System programmer response

Assist the user as necessary.

Module

CMMAKSI

Procedure name

StartUp

EZA9424E **Error on open of *data_set*, sense code = *sense_code*.**

Explanation

CMMAKSI encountered an error in opening the indicated data set. The reason for this error is indicated by the sense code displayed in this message. The data set is not opened.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Use the sense code displayed in this message and the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference to determine the cause of the error.

Module

CMMAKSI

Procedure name

StartUp

EZA9431I **FTP.DATA file not found. Using hardcoded default values.**

Explanation

When the FTP.DATA file is not being used, this message indicates that hardcoded defaults are in effect.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

FTBMVSUB

Procedure name

Exist

EZA9475I **Detected trouble decoding indirected name:**

EZA9476I **Detected trouble decoding name:**

EZA9477I ***undecoded_name***

Explanation

The client was unable to decode a name returned from the Domain Name Server because it exceeded the maximum allowable length. The name is not decoded.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the configuration of the Domain Name Server to determine why it is sending names that are longer than the maximum allowable length.

Module

CMRESOL

Procedure name

GetName

EZA9486I **A**

Explanation

The resource record for which SayRRType was called is an address record. The address record contains the dotted-decimal notation internet address for the domain name identifying the record.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9487I NS

Explanation

The resource record for which SayRRType was called is a name server record. The name server record contains the domain name of a name server for the current zone.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9488I MD

Explanation

The resource record for which SayRRType was called was an MD record. For more information about this experimental record type, see RFC 1034. See Appendix B, “Related protocol specifications,” on page 1065 for information about accessing RFCs.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9489I **MF**

Explanation

The resource record for which SayRRType was called is an MF record.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9490I **CNAME**

Explanation

The resource record for which SayRRType was called is a canonical name record. The canonical name record is used to provide alias or alternative name information for a domain name. The domain name specified in the first field of the record is an alternative to the canonical or real name specified in the data field.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9491I **SOA**

Explanation

The resource record for which SayRRType was called is a start of authority record. The SOA record is unique to a zone and contains administrative details of the zone, including:

- The domain name of the name server responsible for the zone.
- The mail address of the user responsible for the zone.
- The serial number of the zone database, which identifies the current revision of the data.
- The refresh interval, which indicates the length of time, in seconds, you must allow between the refreshing of a database from a remote name server.
- The retry interval, which indicates the length of time, in seconds, you must allow before trying a failed attempt again.
- The expiration time-to-live (TTL), which indicates the maximum time, in seconds, for records to be valid in the zone database.
- The minimum TTL, which indicates the minimum time, in seconds, for records to be valid in the zone database.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9492I **MB**

Explanation

The resource record for which SayRRType was called is a mailbox record. The mailbox record is an experimental record with no RFCs that define specific rules for this data type.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9493I **MG**

Explanation

The resource record for which SayRRType was called is a mail group member record. The mail group member record specifies the mail address of a person belonging to the mail group specified in the domain name field. Mail group records are experimental and have no RFCs that define specific rules for this data type.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9494I **MR**

Explanation

The resource record for which SayRRType was called is a mail rename name record. The mail rename name record specifies a mailbox that is a rename of the mailbox specified in the domain name field. The mail rename name record is an experimental record. There are no RFCs that define specific rules for this data type.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9495I **NULL**

Explanation

The resource record for which SayRRType was called is a null resource record. The null resource record contains any information, providing it is less than 65 535 octets in length. The null resource record is an experimental record type. There are no RFCs that define specific rules for this data type.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9496I **WKS**

Explanation

The resource record for which SayRRType was called is a well-known services record. The well-known services record stores the protocol numbers of multiple services in a single record. See RFC 1060 for more detailed information. See Appendix B, "Related protocol specifications," on page 1065 for information about accessing RFCs.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9497I **PTR**

Explanation

The resource record for which SayRRType was called is a domain name pointer record. The domain name pointer record is mainly used to store data for the `in-addr.arpa` domain, and contains the domain name referenced by an internet address.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9498I **HINFO**

Explanation

The resource record for which SayRRType was called is a host information record. This record type contains a text string specifying the central processing unit (CPU) type and operating system of a node.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9499I **MINFO**

Explanation

The resource record for which SayRRType was called is a mailbox information record. The mailbox information record specifies the mail addresses of the persons responsible for the mail group specified in the domain name field. The mailbox information record is an experimental record type. There are no RFCs that define specific rules for this data type.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9500I **MX**

Explanation

The resource record for which SayRRType was called is a mail exchanger record. The mail exchanger record defines a host that can act as a mail exchange for the domain specified in the domain name field. A mail exchange runs a mail agent that delivers or forwards mail for the domain name specified in the first field.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9501I TXT

Explanation

The resource record for which SayRRType was called is a text string record. The text string record contains descriptive text.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRType

EZA9502I Unknown RR Type

Explanation

SayRRType did not recognize the type of the resource record for which it was called.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Notify the system programmer of the error.

System programmer response

Check the appropriate name server to determine why it is sending incorrect resource records.

Module

CMRESOL

Procedure name

SayRRType

EZA9507I **IN**

Explanation

This message displays the network class. The IN class represents the internet class (most Domain Name Servers support only this class). You can use the wildcard value ANY to match any class. For more information on the network class, see the z/OS Communications Server: IP Configuration Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRClass

EZA9508I **CS**

Explanation

This message displays the network class. The CS class represents the CHAOS class. For more information on the network class of resource records, see the z/OS Communications Server: IP Configuration Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRClass

EZA9509I **CH**

Explanation

This message displays the network class. The CH class represents the CHAOS class. For more information on the network class of resource records, see the z/OS Communications Server: IP Configuration Reference.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRClass

EZA9510I **HS**

Explanation

This message displays the network class. HS represents the Hesoid class. You can use the wildcard ANY to match any class. For more information on the network class, see the z/OS Communications Server: IP User's Guide and Commands.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRClass

EZA9511I **Unknown RR Class (9511)**

Explanation

This message indicates an unknown network class. The valid classes are:

- IN
- CHAOS
- HESOID

For more information on the network class and resource records, see the z/OS Communications Server: IP User's Guide and Commands.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

SayRRClass

EZA9516I **Number of RRs: *number name***

Explanation

This message displays the number of resource records returned in response to a name server query and the name server answering the query.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9517I ***address_class number name***

Explanation

This message displays the address class of a query. You will also receive one or more of the following values, depending on the resource record type:

Value
Indicates

query_type query_class
No responses have been received for this query.

time_to_live
Responses have been received for the query; they will be cached by the name server for the indicated length of time.

host_number
A response has been received for an A type query, the response contains a host address.

cname
A response has been received for a CNAME query. The response contains the canonical name of a host.

mailbox
A response has been received for an MX query. The response contains the address of a mailbox.

name_server
A response has been received for a name server query. The response contains the address of the local name server.

SOA_address
A response has been received for a start of address query. The response contains the address of the start of the table of authority for this zone, and displays the contents of the SOA record.

For more information about these fields, see the z/OS Communications Server: IP User's Guide and Commands.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

varies

Procedure name

varies

EZA9546I **Question**

Explanation

This message displays that the question query option is specified. This option specifies whether to print the question section of the response. The question section contains the original query.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintRR

EZA9547I **Answer**

Explanation

This message displays that the answer option is one of the query options This option specifies whether to print the answer section of the response. The answer section contains the set of all resource records from the name server database that satisfy the query.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintRR

EZA9548I **Authority**

Explanation

Displays that the authority option is specified as one of the query options. This option specifies whether to print the authoritative section of the response. The authoritative section contains the set of all resource records that specify the address of an authoritative name server for the query. This section is used when the name server queried cannot provide an authoritative answer.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintRR

EZA9549I **Additional.**

Explanation

This message displays that the additional query option is specified. This option specifies whether to print the additional section of the response. The additional section contains resource records that have not been explicitly requested, but could be useful.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintRR

EZA9554I *** * * * * Beginning of Message * * * * ***

Explanation

This is the beginning of the message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintMessage

EZA9555I **Query Id: *query id***

Explanation

This message displays the query ID.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintMessage

EZA9556I **Flags: *flags***

Explanation

This message displays the options chosen when querying the name server.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintRR

EZA9557I *** * * * * End of Message * * * * ***

Explanation

This is displayed at the end of the message.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMRESOL

Procedure name

PrintRR

EZA9558I **when row is: *row* col is: *column* curpos is: *row_hexadecimal_sequence, column_hexadecimal_sequence*.**

Explanation

This message displays the hexadecimal sequences for the separate rows and columns displayed.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMSCRNS

Procedure name

initscr

EZA9656E **Invalid row coordinate: *row***

Explanation

TCPIP encountered an incorrect screen row coordinate during a full-screen setup.

Note: The NETSTAT INTERVAL option is valid only for 3278-type terminals.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The NETSTAT program continues.

Operator response

Tell the system programmer the type of terminal on which you encountered the error.

System programmer response

Modify the TCPIP LOGMODE tables to correctly specify the full-screen characteristics of the terminal type on which the user encountered the error.

For information about TCPIP LOGMODE tables, see z/OS Communications Server: IP Configuration Reference.

Module

CMSCRNS

Procedure name

Move

EZA9657E **Invalid column coordinate: *column***

Explanation

TCPIP encountered an incorrect screen column coordinate during a full-screen setup.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

The NETSTAT program halts and exits.

Operator response

Tell the system programmer the type of terminal on which you encountered the error.

Note: The NETSTAT INTERVAL option is valid only for 3278-type terminals.

System programmer response

Modify the TCPIP LOGMODE tables to correctly specify the full-screen characteristics of the terminal type on which the user encountered the error.

For information about TCPIP LOGMODE tables, see z/OS Communications Server: IP Configuration Reference .

Module

CMSCRNS

Procedure name

Move

EZA9689I **DbcsEtoA : First Double Byte Character Invalid *ord(firstchar)***

Explanation

EBCDIC double-byte characters are being converted to ASCII. The first character of a double-byte character, with the specified ordinal value, is not in the valid range for EBCDIC double-byte characters. For translation modes containing both single-byte and double-byte data, the erroneous character will be converted using single-byte EBCDIC to ASCII conversion tables. Data will continue to be converted in this manner until an EBDCIC shift-out character is encountered. If the translation mode uses shift-out and shift-in characters or escape sequences to delimit single-byte and double-byte ASCII data, a shift-in character or escape sequence will be added before the converted erroneous character.

For translation modes that convert pure double-byte data only, the erroneous character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double-byte character. A shift-in character or escape sequence will not be inserted for these modes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid double-byte characters. EBCDIC double-byte characters are enclosed within shift-out `'0E'X` and shift-in `'0F'X` characters.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsEtoA

EZA9690I **DbcsEtoA : Second Double Byte Character Invalid *ord(firstchar) ord(secondchar)***

Explanation

EBCDIC double-byte characters are being converted to ASCII. The second character of a double-byte character, with the specified ordinal value, is not in the valid range for EBCDIC double-byte characters. For translation modes containing both single-byte and double-byte data, the erroneous character and the corresponding first double-byte character will be converted using single byte EBCDIC to ASCII conversion tables. Data will continue to be converted in this manner until an EBDCIC shift-out character is encountered. If the translation mode uses shift-out and shift-in characters or escape sequences to delimit single-byte and double-byte ASCII data, a shift-in character or escape sequence will be added before the converted erroneous characters.

For translation modes that convert pure double-byte data only, the erroneous character and the first double-byte character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double byte character. A shift-in character or escape sequence will not be inserted for these modes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid double-byte characters. EBCDIC double-byte characters are enclosed within shift-out `'0E'X` and shift-in `'0F'X` characters.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsEtoA

EZA9691I **DbcsAtoE : Character after ESCAPE Invalid *ord(ASCIIEscape) ord(nextchar)***

Explanation

ASCII JIS Kanji double-byte characters are being converted to EBCDIC. ASCII JIS Kanji double-byte characters are enclosed within the shift-out escape sequence ESC `$ @` or ESC `$ B`, and the shift-in escape sequence ESC `( B` or ESC `( J`. The character after the ESCAPE, with the specified ordinal value, is not valid for a JIS Kanji shift-out or shift-in sequence. For translation modes containing both single-byte and double-byte data, the erroneous character and the escape character will be converted using single-byte ASCII to EBCDIC conversion tables. Data will continue to be converted in this manner until a valid shift-out escape sequence is encountered. An EBCDIC shift-in character will be inserted before the converted erroneous characters if the error is encountered after a valid double-byte character.

For translation modes that convert pure double-byte data only, the erroneous character and the escape character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double byte character. An EBCDIC shift-in character will not be inserted for these modes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid JIS Kanji double-byte characters and escape sequences for the selected conversion type.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsAtoE

EZA9692I **DbcsAtoE : Character after $ Invalid** ***ord(ASCIIEscape) ord(ASCIIDollar) ord(nextchar)***

Explanation

ASCII JIS Kanji double-byte characters are being converted to EBCDIC. ASCII JIS Kanji double-byte characters are enclosed within the shift-out escape sequence `ESC $ @` or `ESC $ B`, and the shift-in escape sequence `ESC ( B` or `ESC ( J`. The character after the $, with the specified ordinal value, is not valid for a JIS Kanji shift-out sequence. For translation modes containing both single-byte and double-byte data, the erroneous escape sequence will be converted using single byte ASCII to EBCDIC conversion tables. Data will continue to be converted in this manner until a valid Shift-out escape sequence is encountered.

For translation modes that convert pure double-byte data only, the erroneous character and the escape character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double-byte character.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid JIS Kanji double-byte characters and escape sequences for the selected conversion type.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsAtoE

EZA9693I **DbcsAtoE : Character after (Invalid *ord(AsciiEscape) ord(AsciiBraceL) ord(nextchar)***

Explanation

ASCII JIS Kanji double-byte characters are being converted to EBCDIC. ASCII JIS Kanji double-byte characters are enclosed within the shift-out escape sequence `ESC $ @` or `ESC $ B`, and the shift-in escape sequence `ESC ( B` or `ESC ( J`. The character after the (, with the specified ordinal value, is not valid for a JIS Kanji shift-in sequence. For translation modes containing both single-byte and double-byte data, the erroneous escape sequence will be converted using single-byte ASCII to EBCDIC conversion tables. Data will continue to be converted in this manner until a valid shift-out escape sequence is encountered. An EBCDIC shift-in character will be inserted before the erroneous characters.

For translation modes that convert pure double-byte data only, the erroneous character and the escape character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double byte character. An EBCDIC shift-in character will not be inserted for these modes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid JIS Kanji double-byte characters and escape sequences for the selected conversion type.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsAtoE

EZA9694I **DbcsAtoE : Second Double Byte Character Invalid *ord(firstchar) ord(secondchar)***

Explanation

ASCII double-byte characters are being converted to EBCDIC. ASCII double-byte characters are indicated when the first character is in the range for ASCII double-byte characters. The second character of a double-byte character, with the specified ordinal value, is not in the valid range for the second byte of double-byte characters. For translation modes containing both single-byte and double-byte data, the erroneous character and the corresponding first double-byte character will be converted using single byte ASCII to EBCDIC conversion tables. Data will continue to be converted in this manner until a valid shift-out character or valid shift-out escape sequence is encountered. An EBCDIC Shift-in character will be inserted before the converted erroneous characters.

For translation modes that convert pure double-byte data only, the erroneous character and the first double-byte character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double-byte character. An EBCDIC shift-in character will not be inserted for these modes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid ASCII double-byte characters for the selected conversion type.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsAtoE

EZA9695I **DbcsAtoE : First Double Byte Character Invalid *ord(firstchar)***

Explanation

ASCII JIS Kanji double-byte characters are being converted to EBCDIC. ASCII JIS Kanji double-byte characters are enclosed within the shift-out escape sequence ESC `$ @` or ESC `$ B`, and the shift-in escape sequence ESC `( B` or ESC `( J`. The first character of a double-byte character, with the specified ordinal value, is not in the valid range for JIS Kanji double-byte characters. For translation modes containing both single-byte and double-byte data, the erroneous character will be converted using single byte ASCII to EBCDIC conversion tables. Data will continue to be converted in this manner until a shift-out character or escape sequence is encountered. An EBCDIC shift-in character will be added before the converted erroneous character.

For translation modes that convert pure double-byte data only, the erroneous character will be inserted without conversion and the conversion routine will move to the next character assuming it to be a valid first double byte character. An EBCDIC shift-in character will not be inserted for these modes.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid JIS Kanji double-byte characters and escape sequences for the selected conversion type.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsAtoE

EZA9697I **Invalid Data in *data_set* : Loading Stopped.**

Explanation

Binary DBCS translation tables are being loaded from the indicated data set. The data set does not contain data in the required format for DBCS binary translate tables. Loading of the indicated data set stops.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

Processing continues.

Operator response

Configure a valid DBCS binary translate table data set in the search order hierarchy for the required DBCS translation table. See z/OS Communications Server: IP Configuration Reference for information about the loading and customizing of DBCS translation tables.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

ExitError

EZA9698I **DbcsAtoE : Double Byte Character outside Shift-Out/Shift-in delimiters *ord(char)***

Explanation

ASCII double-byte characters are being converted to EBCDIC. ASCII double-byte characters should be enclosed within shift-out/shift-in characters. A character with the indicated ordinal value was found outside the shift-out/shift-in delimiters. The ordinal value was in the range for double-byte characters for this data. The erroneous character will be converted using single-byte ASCII to EBCDIC conversion tables. Data will continue to be converted in this manner until an ASCII shift-out character is encountered.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

Check the contents of the data being transmitted, to see if it contains valid shift-out and shift-in characters for the selected conversion type and DBCS control options.

System programmer response

Assist the user as necessary.

Module

CMDBCSCV PASCAL

Procedure name

DbcsAtoE

EZA9730I **Issue Cp of ‘*command*’**

Explanation

This message is displayed upon invocation of the IssueCP procedure. This function asks CP to invoke the given command and returns the return code and response string that CP provides.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVPMISS

Procedure name

IssueCp

EZA9731I **Write to printer *printer address*: ‘*message*’**

Explanation

The message specified is written to a virtual printer at the address specified.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVPMISS

Procedure name

SioMsg

EZA9736I **Call to Tsch**

Explanation

This message is displayed upon the invocation of the Tsch procedure. This function returns a condition code of 3, indicating that this function is not operational.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCMVMISS

Procedure name

Tsch

EZA9737I **Call to GetSid**

Explanation

This message is displayed upon the invocation of the GetSid function. This function gets the subsystem-identification word corresponding the device address.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCMVMISS

Procedure name

GetSid

EZA9738I **Attempt to send file *data set* to *user***

Explanation

This message is displayed upon invocation of the SendUser procedure. This function simply sends a file to a user.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

TCMVMISS

Procedure name

SendUser

EZA9740I **Initialize program check handler**

Explanation

This message is displayed upon the invocation of the InitProg procedure. This function initializes the program check handler.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVPMISS

Procedure name

InitProg

EZA9742I **NSIREP called.**

Explanation

This message is displayed upon the invocation of the NSIREP procedure.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

MVPMISS

Procedure name

nsirep

EZA9744I **Lqueryoff called.**

Explanation

This message is displayed upon the invocation of the Lqueryoff procedure.

Note: This message displays the virtual machine communication facility (VMCF) function call returned by the function SayVmcFu, which is called to determine the type of a VMCF function call. For more information about VMCF function calls, see the z/OS Communications Server: IP Programmer's Guide and Reference.

System action

TCPIP continues.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

CMMVSUB

Procedure name

Lqueryoff

Chapter 11. EZAIN*xx* messages

EZAIN11I **ALL TCPIP SERVICES FOR PROC *procedure name* ARE AVAILABLE.**

Explanation

TCP/IP has successfully initialized and is ready to process requests.

procedure name is the name of the procedure used to start TCP/IP.

System action

The TCP/IP procedure begins to process user requests.

Operator response

No action is needed.

System programmer response

No action is needed.

Module

EZBTIINI

EZAIN23I **CTRACE IS ACTIVE FOR *jobname* WITH OPTIONS OTHER THAN THE DEFAULT OPTIONS**

Explanation

CTRACE is active with options other than the default options (MINIMUM, INIT, OPCMDS, or OPMSGS). See the information about specifying trace options in the z/OS Communications Server: IP Diagnosis Guide.

jobname is the name of the procedure used to start TCP/IP.

System action

TCP/IP is initialized with CTRACE active.

Operator response

No action is needed.

System programmer response

Running TCP/IP with other than the default CTRACE options can cause a performance degradation. Issue the DISPLAY TRACE,COMP=SYSTCPIP,SUB=(*jobname*) command to display which other CTRACE options are active, and to determine if they need to be set.

Module

EZBTIINI

Procedure name

MAIN

Chapter 12. EZAOP*xx* messages

EZAOP41I **'*mvscommand* TCPIP' COMMAND COMPLETED SUCCESSFULLY**

Explanation

The system successfully processed the DISPLAY or VARY command.

In the message text:

mvscommand
The MVS command verb specified for TCPIP.

System action

Processing continues

System programmer response

No action is needed.

User response

No action is needed.

Module

EZAO2CMD

EZAOP42I **'*mvscommand* TCPIP' SYNTAX ERROR, *description token***

Explanation

A syntax error occurred during the command processing of the specified *mvscommand*. The description and token provide information regarding the error.

In the message text:

mvscommand
The name of the command that was in error.

description
A description of the error. The description can be one of the following:

COMMA (',') EXPECTED
A comma (',') was expected but was not specified.

INVALID PROCNAME
An invalid TCPIP procedure name was specified.

token
The token from the command where the error occurred.

System action

The command request was not completed.

Operator response

Correct the syntax and re-issue the command.

System programmer response

No action is needed.

User response

Correct the syntax and re-issue the command.

Module

EZAO2CMD

EZAOP45I **SPECIFIED TCPIP PROCNAME IS NOT RUNNING: *procname***

Explanation

The TCPIP procedure name in the command was not active.

In the message text:

procname
The TCPIP procedure name for the DISPLAY or VARY command.

System action

The command request is not completed.

System programmer response

No action is needed.

User response

Use the DISPLAY TCPIP command to determine the active TCPIP procedures. Try the command again.

Module

EZAO2CMD

Procedure name

Do_V3R3

EZAOP46I **TCPIP PROCNAME MUST BE SPECIFIED.**

Explanation

More than one TCPIP is currently active and a TCPIP procedure name must be specified in the DISPLAY or VARY command.

System action

The command request is not completed.

System programmer response

No action is needed.

User response

Use the DISPLAY TCPIP command to determine the active TCPIP procedures. Try the command again.

Module

EZAO2CMD

Example

```
d tcpip,,n,dev
EZAOP46I TCPIP PROCNAME MUST BE SPECIFIED.
```

Procedure name

ParseProcName

EZAOP47I *server* HAS NEVER BEEN STARTED

Explanation

This message is issued in response to the D TCPIP or D TCPIP,TELNET display command if the server type has never been started.

In the message text:

server
The type of server that has never been started. The value is either TCPIP or TELNET.

System action

The command request cannot be completed.

Operator response

Try this command again after one or more servers of the type indicated in the message have been started.

System programmer response

No action is needed.

User response

Not applicable.

Problem determination

None.

Source

z/OS Communications Server TCP/IP

Module

EZAO2CMD

Routing code

Not applicable.

Descriptor code

5

Automation

Not applicable.

Example

```
display tcpip
EZAOP47I TCPIP HAS NEVER BEEN STARTED
```

EZAOP49I THERE ARE NO ACTIVE TCPIPS

Explanation

There are no active TCPIP procedures currently running.

System action

The command request cannot be completed.

System programmer response

No action is needed.

User response

Try this command again after one or more TCPIP procedures have been started.

Module

EZAO2CMD

Procedure name

ParseProcName

EZAOP50I TCPIP STATUS REPORT

Explanation

This is a heading for the TCPIP status report in response to the DISPLAY TCPIP command.

System action

The command is accepted.

System programmer response

No action is needed.

User response

No action is needed.

Module

EZAO2DSP

Example

```
d tcpip
EZAOP50I TCPIP STATUS REPORT
COUNT   TCPIP NAME   VERSION    STATUS
-----   ----------   --------   -------------------------
    1   TCPIP        CS V1R8    ACTIVE
*** END TCPIP STATUS REPORT ***
```

Procedure name

OutputTcpStatus

EZAOP60I

TELNET STATUS REPORT

TELNET NAME   VERSION   STATUS

---------------------- --------------   ------------

server_name*   *version*   *status

***** END TELNET STATUS REPORT *****

Explanation

This message is displayed in response to the D TCPIP,TELNET command. The message displays the name, version, and status of each Telnet server that is active now or was previously active.

In the message text:

server_name
The name of the Telnet server.

version
The version and release of Communications Server that was used when the specified server was started.

status
The current state of the Telnet server. Valid states are:

STARTING
The Telnet server is initializing.

ACTIVE
The Telnet server is initialized and ready to support client connections.

TERMINATING
The Telnet server is shutting down and is not supporting connections.

INACTIVE
The Telnet server has shut down, and the procedure has ended.

If the Telnet server has been stopped with the STOP console command, (STOP CMD) is appended to the status.

System action

None.

Operator response

None.

System programmer response

None.

User response

Not applicable.

Problem determination

None.

Source

z/OS Communications Server TCP/IP: TN3270E Telnet Server

Module

EZAO2DSP

Routing code

Not applicable.

Descriptor code

5, 8, 9

Automation

Not applicable.

Example

```
D TCPIP,TELNET
EZAOP60I TELNET STATUS REPORT
TELNET NAME  VERSION    STATUS
-----------  --------   --------------------------------
TELNET       CS V1R13   ACTIVE
TELNET5      CS V1R13   INACTIVE (STOP CMD)
TELNET4      CS V1R13   INACTIVE (STOP CMD)
*** END TELNET STATUS REPORT ***
```

Appendix A. Additional information about remote procedure call (RPC) messages

This appendix contains information about RPCLIB and RPCINFO messages.

RPCLIB messages

To solve remote procedure call (RPC) errors, begin by doing the following actions:

- Check the system console to make sure that PORTMAP and TCPIP are running.
- Check that sufficient storage is available for tasks. The required heap size varies, but could also be the cause of the problem.
- Compress data sets periodically.

See the z/OS Communications Server: IP Programmer's Guide and Reference for information about basic RPC procedures and the requirements for RPC parameters. From this reference, you can determine the missing or incorrect information that caused the error. This reference also lists the tcperror and perror codes, and many of the procedures described in this chapter. Other procedures are contained in the source code and are made up of the basic procedures.

Some RPC errors are caused by C language coding problems. See the following common C language coding errors:

- Array subscripts out of bounds.
- Bad pointers in the user source code. Pointers to incorrect addresses can cause a number of different errors, such as protection violations and overwritten storage.
- Not freeing storage at the end of the job.
- Overwriting data or program areas, (can be an unterminated string). Strings require a null (X'00' or \0) in the last position.
- Pointer misspecification, such as *p, p, &p, in the user source code.
- Unending loops that generate storage usage.

RPCINFO messages

For RPCINFO errors, PING and NETSTAT are important tools. You can use RPCINFO to check procedures registered on local and remote PORTMAPs. PING verifies that local and remote hosts are running with TCPIP. NETSTAT provides the status of socket connections and many other conditions.

Appendix B. Related protocol specifications

This appendix lists the related protocol specifications (RFCs) for TCP/IP. The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. Some of these protocols are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement these particular functions or protocols. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

RFCs are available at http://www.rfc-editor.org/rfc.html.

Draft RFCs that have been implemented in this and previous Communications Server releases are listed at the end of this topic.

Many features of TCP/IP Services are based on the following RFCs:

RFC
    Title and Author

RFC 652
    Telnet output carriage-return disposition option D. Crocker

RFC 653
    Telnet output horizontal tabstops option D. Crocker

RFC 654
    Telnet output horizontal tab disposition option D. Crocker

RFC 655
    Telnet output formfeed disposition option D. Crocker

RFC 657
    Telnet output vertical tab disposition option D. Crocker

RFC 658
    Telnet output linefeed disposition D. Crocker

RFC 698
    Telnet extended ASCII option T. Mock

RFC 726
    Remote Controlled Transmission and Echoing Telnet option J. Postel, D. Crocker

RFC 727
    Telnet logout option M.R. Crispin

RFC 732
    Telnet Data Entry Terminal option J.D. Day

RFC 733
    Standard for the format of ARPA network text messages D. Crocker, J. Vittal, K.T. Pogran, D.A. Henderson

RFC 734
    SUPDUP Protocol M.R. Crispin

RFC 735
    Revised Telnet byte macro option D. Crocker, R.H. Gumpertz

RFC 736
    Telnet SUPDUP option M.R. Crispin

RFC 749
    Telnet SUPDUP—Output option B. Greenberg

RFC 765
    File Transfer Protocol specification J. Postel

RFC 768
User Datagram Protocol J. Postel

RFC 779
Telnet send-location option E. Killian

RFC 791
Internet Protocol J. Postel

RFC 792
Internet Control Message Protocol J. Postel

RFC 793
Transmission Control Protocol J. Postel

RFC 820
Assigned numbers J. Postel

RFC 823
DARPA Internet gateway R. Hinden, A. Sheltzer

RFC 826
Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware D. Plummer

RFC 854
Telnet Protocol Specification J. Postel, J. Reynolds

RFC 855
Telnet Option Specification J. Postel, J. Reynolds

RFC 856
Telnet Binary Transmission J. Postel, J. Reynolds

RFC 857
Telnet Echo Option J. Postel, J. Reynolds

RFC 858
Telnet Suppress Go Ahead Option J. Postel, J. Reynolds

RFC 859
Telnet Status Option J. Postel, J. Reynolds

RFC 860
Telnet Timing Mark Option J. Postel, J. Reynolds

RFC 861
Telnet Extended Options: List Option J. Postel, J. Reynolds

RFC 862
Echo Protocol J. Postel

RFC 863
Discard Protocol J. Postel

RFC 864
Character Generator Protocol J. Postel

RFC 865
Quote of the Day Protocol J. Postel

RFC 868
Time Protocol J. Postel, K. Harrenstien

RFC 877
Standard for the transmission of IP datagrams over public data networks J.T. Korb

RFC 883
Domain names: Implementation specification P.V. Mockapetris

RFC 884
Telnet terminal type option M. Solomon, E. Wimmers

RFC 885
Telnet end of record option J. Postel

RFC 894
Standard for the transmission of IP datagrams over Ethernet networks C. Hornig

RFC 896
Congestion control in IP/TCP internetworks J. Nagle

RFC 903
Reverse Address Resolution Protocol R. Finlayson, T. Mann, J. Mogul, M. Theimer

RFC 904
Exterior Gateway Protocol formal specification D. Mills

RFC 919
Broadcasting Internet Datagrams J. Mogul

RFC 922
Broadcasting Internet datagrams in the presence of subnets J. Mogul

RFC 927
TACACS user identification Telnet option B.A. Anderson

RFC 933
Output marking Telnet option S. Silverman

RFC 946
Telnet terminal location number option R. Nedved

RFC 950
Internet Standard Subnetting Procedure J. Mogul, J. Postel

RFC 952
DoD Internet host table specification K. Harrenstien, M. Stahl, E. Feinler

RFC 959
File Transfer Protocol J. Postel, J.K. Reynolds

RFC 961
Official ARPA-Internet protocols J.K. Reynolds, J. Postel

RFC 974
Mail routing and the domain system C. Partridge

RFC 1001
Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force

RFC 1002
Protocol Standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force

RFC 1006
ISO transport services on top of the TCP: Version 3 M.T. Rose, D.E. Cass

RFC 1009
Requirements for Internet gateways R. Braden, J. Postel

RFC 1011
Official Internet protocols J. Reynolds, J. Postel

RFC 1013
X Window System Protocol, version 11: Alpha update April 1987 R. Scheifler

RFC 1014
XDR: External Data Representation standard Sun Microsystems

RFC 1027
Using ARP to implement transparent subnet gateways S. Carl-Mitchell, J. Quarterman

RFC 1032
Domain administrators guide M. Stahl

RFC 1033
Domain administrators operations guide M. Lottor

RFC 1034
Domain names—concepts and facilities P.V. Mockapetris

RFC 1035
Domain names—implementation and specification P.V. Mockapetris

RFC 1038
Draft revised IP security option M. St. Johns

RFC 1041
Telnet 3270 regime option Y. Rekhter

RFC 1042
Standard for the transmission of IP datagrams over IEEE 802 networks J. Postel, J. Reynolds

RFC 1043
Telnet Data Entry Terminal option: DODIIS implementation A. Yasuda, T. Thompson

RFC 1044
Internet Protocol on Network System's HYPERchannel: Protocol specification K. Hardwick, J. Lekashman

RFC 1053
Telnet X.3 PAD option S. Levy, T. Jacobson

RFC 1055
Nonstandard for transmission of IP datagrams over serial lines: SLIP J. Romkey

RFC 1057
RPC: Remote Procedure Call Protocol Specification: Version 2 Sun Microsystems

RFC 1058
Routing Information Protocol C. Hedrick

RFC 1060
Assigned numbers J. Reynolds, J. Postel

RFC 1067
Simple Network Management Protocol J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin

RFC 1071
Computing the Internet checksum R.T. Braden, D.A. Borman, C. Partridge

RFC 1072
TCP extensions for long-delay paths V. Jacobson, R.T. Braden

RFC 1073
Telnet window size option D. Waitzman

RFC 1079
Telnet terminal speed option C. Hedrick

RFC 1085
ISO presentation services on top of TCP/IP based internets M.T. Rose

RFC 1091
Telnet terminal-type option J. VanBokkelen

RFC 1094
NFS: Network File System Protocol specification Sun Microsystems

RFC 1096
Telnet X display location option G. Marcy

RFC 1101
DNS encoding of network names and other types P. Mockapetris

RFC 1112
Host extensions for IP multicasting S.E. Deering

RFC 1113
Privacy enhancement for Internet electronic mail: Part I — message encipherment and authentication procedures J. Linn

RFC 1118
Hitchhikers Guide to the Internet E. Krol

RFC 1122
Requirements for Internet Hosts—Communication Layers R. Braden, Ed.

RFC 1123
Requirements for Internet Hosts—Application and Support R. Braden, Ed.

RFC 1146
TCP alternate checksum options J. Zweig, C. Partridge

RFC 1155
Structure and identification of management information for TCP/IP-based internets M. Rose, K. McCloghrie

RFC 1156
Management Information Base for network management of TCP/IP-based internets K. McCloghrie, M. Rose

RFC 1157
Simple Network Management Protocol (SNMP) J. Case, M. Fedor, M. Schoffstall, J. Davin

RFC 1158
Management Information Base for network management of TCP/IP-based internets: MIB-II M. Rose

RFC 1166
Internet numbers S. Kirkpatrick, M.K. Stahl, M. Recker

RFC 1179
Line printer daemon protocol L. McLaughlin

RFC 1180
TCP/IP tutorial T. Socolofsky, C. Kale

RFC 1183
New DNS RR Definitions C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris

RFC 1184
Telnet Linemode Option D. Borman

RFC 1186
MD4 Message Digest Algorithm R.L. Rivest

RFC 1187
Bulk Table Retrieval with the SNMP M. Rose, K. McCloghrie, J. Davin

RFC 1188
Proposed Standard for the Transmission of IP Datagrams over FDDI Networks D. Katz

RFC 1190
Experimental Internet Stream Protocol: Version 2 (ST-II) C. Topolcic

RFC 1191
Path MTU discovery J. Mogul, S. Deering

RFC 1198
FYI on the X window system R. Scheifler

RFC 1207
FYI on Questions and Answers: Answers to commonly asked "experienced Internet user" questions G. Malkin, A. Marine, J. Reynolds

RFC 1208
Glossary of networking terms O. Jacobsen, D. Lynch

RFC 1213
Management Information Base for Network Management of TCP/IP-based internets: MIB-II K. McCloghrie, M.T. Rose

RFC 1215
Convention for defining traps for use with the SNMP M. Rose

RFC 1227
SNMP MUX protocol and MIB M.T. Rose

RFC 1228
SNMP-DPI: Simple Network Management Protocol Distributed Program Interface G. Carpenter, B. Wijnen

RFC 1229
Extensions to the generic-interface MIB K. McCloghrie

RFC 1230
IEEE 802.4 Token Bus MIB K. McCloghrie, R. Fox

RFC 1231
IEEE 802.5 Token Ring MIB K. McCloghrie, R. Fox, E. Decker

RFC 1236
IP to X.121 address mapping for DDN L. Morales, P. Hasse

RFC 1256
ICMP Router Discovery Messages S. Deering, Ed.

RFC 1267
Border Gateway Protocol 3 (BGP-3) K. Lougheed, Y. Rekhter

RFC 1268
Application of the Border Gateway Protocol in the Internet Y. Rekhter, P. Gross

RFC 1269
Definitions of Managed Objects for the Border Gateway Protocol: Version 3 S. Willis, J. Burruss

RFC 1270
SNMP Communications Services F. Kastenholz, ed.

RFC 1285
FDDI Management Information Base J. Case

RFC 1315
Management Information Base for Frame Relay DTEs C. Brown, F. Baker, C. Carvalho

RFC 1321
The MD5 Message-Digest Algorithm R. Rivest

RFC 1323
TCP Extensions for High Performance V. Jacobson, R. Braden, D. Borman

RFC 1325
FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions G. Malkin, A. Marine

RFC 1327
Mapping between X.400 (1988)/ISO 10021 and RFC 822 S. Hardcastle-Kille

RFC 1340
Assigned Numbers J. Reynolds, J. Postel

RFC 1344
Implications of MIME for Internet Mail Gateways N. Bornstein

RFC 1349
Type of Service in the Internet Protocol Suite P. Almquist

RFC 1351
SNMP Administrative Model J. Davin, J. Galvin, K. McCloghrie

RFC 1352
SNMP Security Protocols J. Galvin, K. McCloghrie, J. Davin

RFC 1353
Definitions of Managed Objects for Administration of SNMP Parties K. McCloghrie, J. Davin, J. Galvin

RFC 1354
IP Forwarding Table MIB F. Baker

RFC 1356
Multiprotocol Interconnect® on X.25 and ISDN in the Packet Mode A. Malis, D. Robinson, R. Ullmann

RFC 1358
Charter of the Internet Architecture Board (IAB) L. Chapin

RFC 1363
A Proposed Flow Specification C. Partridge

RFC 1368
Definition of Managed Objects for IEEE 802.3 Repeater Devices D. McMaster, K. McCloghrie

RFC 1372
Telnet Remote Flow Control Option C. L. Hedrick, D. Borman

RFC 1374
IP and ARP on HIPPI J. Renwick, A. Nicholson

RFC 1381
SNMP MIB Extension for X.25 LAPB D. Throop, F. Baker

RFC 1382
SNMP MIB Extension for the X.25 Packet Layer D. Throop

RFC 1387
RIP Version 2 Protocol Analysis G. Malkin

RFC 1388
RIP Version 2 Carrying Additional Information G. Malkin

RFC 1389
RIP Version 2 MIB Extensions G. Malkin, F. Baker

RFC 1390
Transmission of IP and ARP over FDDI Networks D. Katz

RFC 1393
Traceroute Using an IP Option G. Malkin

RFC 1398
Definitions of Managed Objects for the Ethernet-Like Interface Types F. Kastenholz

RFC 1408
Telnet Environment Option D. Borman, Ed.

RFC 1413
Identification Protocol M. St. Johns

RFC 1416
Telnet Authentication Option D. Borman, ed.

RFC 1420
SNMP over IPX S. Bostock

RFC 1428
Transition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME G. Vaudreuil

RFC 1442
Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1443
Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1445
Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2) J. Galvin, K. McCloghrie

RFC 1447
Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2) K. McCloghrie, J. Galvin

RFC 1448
Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1464
Using the Domain Name System to Store Arbitrary String Attributes R. Rosenbaum

RFC 1469
IP Multicast over Token-Ring Local Area Networks T. Pusateri

RFC 1483
Multiprotocol Encapsulation over ATM Adaptation Layer 5 Juha Heinanen

RFC 1514
Host Resources MIB P. Grillo, S. Waldbusser

RFC 1516
Definitions of Managed Objects for IEEE 802.3 Repeater Devices D. McMaster, K. McCloghrie

RFC 1521
MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies N. Borenstein, N. Freed

RFC 1535
A Security Problem and Proposed Correction With Widely Deployed DNS Software E. Gavron

RFC 1536
Common DNS Implementation Errors and Suggested Fixes A. Kumar, J. Postel, C. Neuman, P. Danzig, S. Miller

RFC 1537
Common DNS Data File Configuration Errors P. Beertema

RFC 1540
Internet Official Protocol Standards J. Postel

RFC 1571
Telnet Environment Option Interoperability Issues D. Borman

RFC 1572
Telnet Environment Option S. Alexander

RFC 1573
Evolution of the Interfaces Group of MIB-II K. McCloghrie, F. Kastenholz

RFC 1577
Classical IP and ARP over ATM M. Laubach

RFC 1583
OSPF Version 2 J. Moy

RFC 1591
Domain Name System Structure and Delegation J. Postel

RFC 1592
Simple Network Management Protocol Distributed Protocol Interface Version 2.0 B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

RFC 1594
FYI on Questions and Answers— Answers to Commonly Asked "New Internet User" Questions A. Marine, J. Reynolds, G. Malkin

RFC 1644
T/TCP — TCP Extensions for Transactions Functional Specification R. Braden

RFC 1646
TN3270 Extensions for LUname and Printer Selection C. Graves, T. Butts, M. Angel

RFC 1647
TN3270 Enhancements B. Kelly

RFC 1652
SMTP Service Extension for 8bit-MIMEtransport J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker

RFC 1664
Using the Internet DNS to Distribute RFC1327 Mail Address Mapping Tables C. Allochio, A. Bonito, B. Cole, S. Giordano, R. Hagens

RFC 1693
An Extension to TCP: Partial Order Service T. Connolly, P. Amer, P. Conrad

RFC 1695
Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2 M. Ahmed, K. Tesink

RFC 1701
Generic Routing Encapsulation (GRE) S. Hanks, T. Li, D. Farinacci, P. Traina

RFC 1702
Generic Routing Encapsulation over IPv4 networks S. Hanks, T. Li, D. Farinacci, P. Traina

RFC 1706
DNS NSAP Resource Records B. Manning, R. Colella

RFC 1712
DNS Encoding of Geographical Location C. Farrell, M. Schulze, S. Pleitner D. Baldoni

RFC 1713
Tools for DNS debugging A. Romao

RFC 1723
RIP Version 2—Carrying Additional Information G. Malkin

RFC 1752
The Recommendation for the IP Next Generation Protocol S. Bradner, A. Mankin

RFC 1766
Tags for the Identification of Languages H. Alvestrand

RFC 1771
A Border Gateway Protocol 4 (BGP-4) Y. Rekhter, T. Li

RFC 1794
DNS Support for Load Balancing T. Brisco

RFC 1819
Internet Stream Protocol Version 2 (ST2) Protocol Specification—Version ST2+ L. Delgrossi, L. Berger Eds.

RFC 1826
IP Authentication Header R. Atkinson

RFC 1828
IP Authentication using Keyed MD5 P. Metzger, W. Simpson

RFC 1829
The ESP DES-CBC Transform P. Karn, P. Metzger, W. Simpson

RFC 1830
SMTP Service Extensions for Transmission of Large and Binary MIME Messages G. Vaudreuil

RFC 1831
RPC: Remote Procedure Call Protocol Specification Version 2 R. Srinivasan

RFC 1832
XDR: External Data Representation Standard R. Srinivasan

RFC 1833
Binding Protocols for ONC RPC Version 2 R. Srinivasan

RFC 1850
OSPF Version 2 Management Information Base F. Baker, R. Coltun

RFC 1854
SMTP Service Extension for Command Pipelining N. Freed

RFC 1869
SMTP Service Extensions J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker

RFC 1870
SMTP Service Extension for Message Size Declaration J. Klensin, N. Freed, K. Moore

RFC 1876
A Means for Expressing Location Information in the Domain Name System C. Davis, P. Vixie, T. Goodwin, I. Dickinson

RFC 1883
Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden

RFC 1884
IP Version 6 Addressing Architecture R. Hinden, S. Deering, Eds.

RFC 1886
DNS Extensions to support IP version 6 S. Thomson, C. Huitema

RFC 1888
OSI NSAPs and IPv6 J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd

RFC 1891
SMTP Service Extension for Delivery Status Notifications K. Moore

RFC 1892
The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages G. Vaudreuil

RFC 1894
*An Extensible Message Format for Delivery Status Notifications*K. Moore, G. Vaudreuil

RFC 1901
Introduction to Community-based SNMPv2 J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1902
Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1903
Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1904
Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1905
Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1906
Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1907
Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1908
Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1912
Common DNS Operational and Configuration Errors D. Barr

RFC 1918
Address Allocation for Private Internets Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear

RFC 1928
SOCKS Protocol Version 5 M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones

RFC 1930
Guidelines for creation, selection, and registration of an Autonomous System (AS) J. Hawkinson, T. Bates

RFC 1939
Post Office Protocol-Version 3 J. Myers, M. Rose

RFC 1981
Path MTU Discovery for IP version 6 J. McCann, S. Deering, J. Mogul

RFC 1982
Serial Number Arithmetic R. Elz, R. Bush

RFC 1985
SMTP Service Extension for Remote Message Queue Starting J. De Winter

RFC 1995
Incremental Zone Transfer in DNS M. Ohta

RFC 1996
A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY) P. Vixie

RFC 2010
Operational Criteria for Root Name Servers B. Manning, P. Vixie

RFC 2011
SNMPv2 Management Information Base for the Internet Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2012
SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2013
SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2018
TCP Selective Acknowledgement Options M. Mathis, J. Mahdavi, S. Floyd, A. Romanow

RFC 2026
The Internet Standards Process — Revision 3 S. Bradner

RFC 2030
Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI D. Mills

RFC 2033
Local Mail Transfer Protocol J. Myers

RFC 2034
*SMTP Service Extension for Returning Enhanced Error Codes*N. Freed

RFC 2040
*The RC5, RC5–CBC, RC-5–CBC-Pad, and RC5–CTS Algorithms*R. Baldwin, R. Rivest

RFC 2045
Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies N. Freed, N. Borenstein

RFC 2052
A DNS RR for specifying the location of services (DNS SRV) A. Gulbrandsen, P. Vixie

RFC 2065
Domain Name System Security Extensions D. Eastlake 3rd, C. Kaufman

RFC 2066
TELNET CHARSET Option R. Gellens

RFC 2080
RIPng for IPv6 G. Malkin, R. Minnear

RFC 2096
IP Forwarding Table MIB F. Baker

RFC 2104
HMAC: Keyed-Hashing for Message Authentication H. Krawczyk, M. Bellare, R. Canetti

RFC 2119
Keywords for use in RFCs to Indicate Requirement Levels S. Bradner

RFC 2133
Basic Socket Interface Extensions for IPv6 R. Gilligan, S. Thomson, J. Bound, W. Stevens

RFC 2136
Dynamic Updates in the Domain Name System (DNS UPDATE) P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound

RFC 2137
Secure Domain Name System Dynamic Update D. Eastlake 3rd

RFC 2163
Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM) C. Allocchio

RFC 2168
Resolution of Uniform Resource Identifiers using the Domain Name System R. Daniel, M. Mealling

RFC 2178
OSPF Version 2 J. Moy

RFC 2181
Clarifications to the DNS Specification R. Elz, R. Bush

RFC 2205
Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin

RFC 2210
The Use of RSVP with IETF Integrated Services J. Wroclawski

RFC 2211
Specification of the Controlled-Load Network Element Service J. Wroclawski

RFC 2212
Specification of Guaranteed Quality of Service S. Shenker, C. Partridge, R. Guerin

RFC 2215
General Characterization Parameters for Integrated Service Network Elements S. Shenker, J. Wroclawski

RFC 2217
Telnet Com Port Control Option G. Clarke

RFC 2219
Use of DNS Aliases for Network Services M. Hamilton, R. Wright

RFC 2228
FTP Security Extensions M. Horowitz, S. Lunt

RFC 2230
Key Exchange Delegation Record for the DNS R. Atkinson

RFC 2233
The Interfaces Group MIB using SMIv2 K. McCloghrie, F. Kastenholz

RFC 2240
A Legal Basis for Domain Name Allocation O. Vaughn

RFC 2246
The TLS Protocol Version 1.0 T. Dierks, C. Allen

RFC 2251
Lightweight Directory Access Protocol (v3) M. Wahl, T. Howes, S. Kille

RFC 2253
Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names M. Wahl, S. Kille, T. Howes

RFC 2254
The String Representation of LDAP Search Filters T. Howes

RFC 2261
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 2262
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 2271
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 2273
SNMPv3 Applications D. Levi, P. Meyer, B. Stewartz

RFC 2274
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 2275
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 2279
UTF-8, a transformation format of ISO 10646 F. Yergeau

RFC 2292
Advanced Sockets API for IPv6 W. Stevens, M. Thomas

RFC 2308
Negative Caching of DNS Queries (DNS NCACHE) M. Andrews

RFC 2317
Classless IN-ADDR.ARPA delegation H. Eidnes, G. de Groot, P. Vixie

RFC 2320
Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB) M. Greene, J. Luciani, K. White, T. Kuo

RFC 2328
OSPF Version 2 J. Moy

RFC 2345
Domain Names and Company Name Retrieval J. Klensin, T. Wolf, G. Oglesby

RFC 2352
A Convention for Using Legal Names as Domain Names O. Vaughn

RFC 2355
TN3270 Enhancements B. Kelly

RFC 2358
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson

RFC 2373
IP Version 6 Addressing Architecture R. Hinden, S. Deering

RFC 2374
An IPv6 Aggregatable Global Unicast Address Format R. Hinden, M. O'Dell, S. Deering

RFC 2375
IPv6 Multicast Address Assignments R. Hinden, S. Deering

RFC 2385
Protection of BGP Sessions via the TCP MD5 Signature Option A. Hefferman

RFC 2389
Feature negotiation mechanism for the File Transfer Protocol P. Hethmon, R. Elz

RFC 2401
Security Architecture for Internet Protocol S. Kent, R. Atkinson

RFC 2402
IP Authentication Header S. Kent, R. Atkinson

RFC 2403
The Use of HMAC-MD5–96 within ESP and AH C. Madson, R. Glenn

RFC 2404
The Use of HMAC-SHA–1–96 within ESP and AH C. Madson, R. Glenn

RFC 2405
The ESP DES-CBC Cipher Algorithm With Explicit IV C. Madson, N. Doraswamy

RFC 2406
IP Encapsulating Security Payload (ESP) S. Kent, R. Atkinson

RFC 2407
*The Internet IP Security Domain of Interpretation for ISAKMP*D. Piper

RFC 2408
Internet Security Association and Key Management Protocol (ISAKMP) D. Maughan, M. Schertler, M. Schneider, J. Turner

RFC 2409
The Internet Key Exchange (IKE) D. Harkins, D. Carrel

RFC 2410
The NULL Encryption Algorithm and Its Use With IPsec R. Glenn, S. Kent,

RFC 2428
FTP Extensions for IPv6 and NATs M. Allman, S. Ostermann, C. Metz

RFC 2445
Internet Calendaring and Scheduling Core Object Specification (iCalendar) F. Dawson, D. Stenerson

RFC 2459
Internet X.509 Public Key Infrastructure Certificate and CRL Profile R. Housley, W. Ford, W. Polk, D. Solo

RFC 2460
Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden

RFC 2461
Neighbor Discovery for IP Version 6 (IPv6) T. Narten, E. Nordmark, W. Simpson

RFC 2462
IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten

RFC 2463
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

RFC 2464
Transmission of IPv6 Packets over Ethernet Networks M. Crawford

RFC 2466
Management Information Base for IP Version 6: ICMPv6 Group D. Haskin, S. Onishi

RFC 2476
Message Submission R. Gellens, J. Klensin

RFC 2487
SMTP Service Extension for Secure SMTP over TLS P. Hoffman

RFC 2505
Anti-Spam Recommendations for SMTP MTAs G. Lindberg

RFC 2523
Photuris: Extended Schemes and Attributes P. Karn, W. Simpson

RFC 2535
Domain Name System Security Extensions D. Eastlake 3rd

RFC 2538
Storing Certificates in the Domain Name System (DNS) D. Eastlake 3rd, O. Gudmundsson

RFC 2539
Storage of Diffie-Hellman Keys in the Domain Name System (DNS) D. Eastlake 3rd

RFC 2540
Detached Domain Name System (DNS) Information D. Eastlake 3rd

RFC 2554
SMTP Service Extension for Authentication J. Myers

RFC 2570
Introduction to Version 3 of the Internet-standard Network Management Framework J. Case, R. Mundy, D. Partain, B. Stewart

RFC 2571
An Architecture for Describing SNMP Management Frameworks B. Wijnen, D. Harrington, R. Presuhn

RFC 2572
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 2573
SNMP Applications D. Levi, P. Meyer, B. Stewart

RFC 2574
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 2575
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 2576
Co-Existence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework R. Frye, D. Levi, S. Routhier, B. Wijnen

RFC 2578
Structure of Management Information Version 2 (SMIv2) K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2579
Textual Conventions for SMIv2 K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2580
Conformance Statements for SMIv2 K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2581
TCP Congestion Control M. Allman, V. Paxson, W. Stevens

RFC 2583
Guidelines for Next Hop Client (NHC) Developers R. Carlson, L. Winkler

RFC 2591
Definitions of Managed Objects for Scheduling Management Operations D. Levi, J. Schoenwaelder

RFC 2625
IP and ARP over Fibre Channel M. Rajagopal, R. Bhagwat, W. Rickard

RFC 2635
Don't SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam)* S. Hambridge, A. Lunde

RFC 2637
Point-to-Point Tunneling Protocol K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, G. Zorn

RFC 2640
Internationalization of the File Transfer Protocol B. Curtin

RFC 2665
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson

RFC 2671
Extension Mechanisms for DNS (EDNS0) P. Vixie

RFC 2672
Non-Terminal DNS Name Redirection M. Crawford

RFC 2675
IPv6 Jumbograms D. Borman, S. Deering, R. Hinden

RFC 2710
Multicast Listener Discovery (MLD) for IPv6 S. Deering, W. Fenner, B. Haberman

RFC 2711
IPv6 Router Alert Option C. Partridge, A. Jackson

RFC 2740
OSPF for IPv6 R. Coltun, D. Ferguson, J. Moy

RFC 2753
A Framework for Policy-based Admission Control R. Yavatkar, D. Pendarakis, R. Guerin

RFC 2782
A DNS RR for specifying the location of services (DNS SRV) A. Gubrandsen, P. Vixix, L. Esibov

RFC 2821
Simple Mail Transfer Protocol J. Klensin, Ed.

RFC 2822
Internet Message Format P. Resnick, Ed.

RFC 2840
TELNET KERMIT OPTION J. Altman, F. da Cruz

RFC 2845
Secret Key Transaction Authentication for DNS (TSIG) P. Vixie, O. Gudmundsson, D. Eastlake 3rd, B. Wellington

RFC 2851
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 2852
Deliver By SMTP Service Extension D. Newman

RFC 2874
DNS Extensions to Support IPv6 Address Aggregation and Renumbering M. Crawford, C. Huitema

RFC 2915
The Naming Authority Pointer (NAPTR) DNS Resource Record M. Mealling, R. Daniel

RFC 2920
SMTP Service Extension for Command Pipelining N. Freed

RFC 2930
Secret Key Establishment for DNS (TKEY RR) D. Eastlake, 3rd

RFC 2941
Telnet Authentication Option T. Ts'o, ed., J. Altman

RFC 2942
Telnet Authentication: Kerberos Version 5 T. Ts'o

RFC 2946
Telnet Data Encryption Option T. Ts'o

RFC 2952
Telnet Encryption: DES 64 bit Cipher Feedback T. Ts'o

RFC 2953

Telnet Encryption: DES 64 bit Output Feedback T. Ts'o

RFC 2992

Analysis of an Equal-Cost Multi-Path Algorithm C. Hopps

RFC 3019

IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol B. Haberman, R. Worzella

RFC 3060

Policy Core Information Model—Version 1 Specification B. Moore, E. Ellesson, J. Strassner, A. Westerinen

RFC 3152

Delegation of IP6.ARPA R. Bush

RFC 3164

The BSD Syslog Protocol C. Lonvick

RFC 3207

SMTP Service Extension for Secure SMTP over Transport Layer Security P. Hoffman

RFC 3226

DNSSEC and IPv6 A6 aware server/resolver message size requirements O. Gudmundsson

RFC 3291

Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 3363

Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System R. Bush, A. Durand, B. Fink, O. Gudmundsson, T. Hain

RFC 3376

Internet Group Management Protocol, Version 3 B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan

RFC 3390

Increasing TCP's Initial Window M. Allman, S. Floyd, C. Partridge

RFC 3410

Introduction and Applicability Statements for Internet-Standard Management Framework J. Case, R. Mundy, D. Partain, B. Stewart

RFC 3411

An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 3412

Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 3413

Simple Network Management Protocol (SNMP) Applications D. Levi, P. Meyer, B. Stewart

RFC 3414

User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 3415

View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 3416

Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP) R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3417
Transport Mappings for the Simple Network Management Protocol (SNMP) R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3418
Management Information Base (MIB) for the Simple Network Management Protocol (SNMP) R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3419
Textual Conventions for Transport Addresses M. Daniele, J. Schoenwaelder

RFC 3484
Default Address Selection for Internet Protocol version 6 (IPv6) R. Draves

RFC 3493
Basic Socket Interface Extensions for IPv6 R. Gilligan, S. Thomson, J. Bound, J. McCann, W. Stevens

RFC 3513
Internet Protocol Version 6 (IPv6) Addressing Architecture R. Hinden, S. Deering

RFC 3526
More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE) T. Kivinen, M. Kojo

RFC 3542
Advanced Sockets Application Programming Interface (API) for IPv6 W. Richard Stevens, M. Thomas, E. Nordmark, T. Jinmei

RFC 3566
The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec S. Frankel, H. Herbert

RFC 3569
An Overview of Source-Specific Multicast (SSM) S. Bhattacharyya, Ed.

RFC 3584
Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework R. Frye, D. Levi, S. Routhier, B. Wijnen

RFC 3602
The AES-CBC Cipher Algorithm and Its Use with IPsec S. Frankel, R. Glenn, S. Kelly

RFC 3629
UTF-8, a transformation format of ISO 10646 R. Kermode, C. Vicisano

RFC 3658
Delegation Signer (DS) Resource Record (RR) O. Gudmundsson

RFC 3678
Socket Interface Extensions for Multicast Source Filters D. Thaler, B. Fenner, B. Quinn

RFC 3715
IPsec-Network Address Translation (NAT) Compatibility Requirements B. Aboba, W. Dixon

RFC 3810
Multicast Listener Discovery Version 2 (MLDv2) for IPv6 R. Vida, Ed., L. Costa, Ed.

RFC 3826
The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model U. Blumenthal, F. Maino, K McCloghrie.

RFC 3947
Negotiation of NAT-Traversal in the IKE T. Kivinen, B. Swander, A. Huttunen, V. Volpe

RFC 3948
UDP Encapsulation of IPsec ESP Packets A. Huttunen, B. Swander, V. Volpe, L. DiBurro, M. Stenberg

RFC 4001
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 4007
IPv6 Scoped Address Architecture S. Deering, B. Haberman, T. Jinmei, E. Nordmark, B. Zill

RFC 4022
Management Information Base for the Transmission Control Protocol (TCP) R. Raghunarayan

RFC 4106
The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) J. Viega, D. McGrew

RFC 4109
Algorithms for Internet Key Exchange version 1 (IKEv1) P. Hoffman

RFC 4113
Management Information Base for the User Datagram Protocol (UDP) B. Fenner, J. Flick

RFC 4191
Default Router Preferences and More-Specific Routes R. Draves, D. Thaler

RFC 4217
Securing FTP with TLS P. Ford-Hutchinson

RFC 4292
IP Forwarding Table MIB B. Haberman

RFC 4293
Management Information Base for the Internet Protocol (IP) S. Routhier

RFC 4301
Security Architecture for the Internet Protocol S. Kent, K. Seo

RFC 4302
IP Authentication Header S. Kent

RFC 4303
IP Encapsulating Security Payload (ESP) S. Kent

RFC 4304
Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP) S. Kent

RFC 4307
Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2) J. Schiller

RFC 4308
Cryptographic Suites for IPsec P. Hoffman

RFC 4434
The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol P. Hoffman

RFC 4443
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

RFC 4552
Authentication/Confidentiality for OSPFv3 M. Gupta, N. Melam

RFC 4678
Server/Application State Protocol v1 A. Bivens

RFC 4753
ECP Groups for IKE and IKEv2 D. Fu, J. Solinas

RFC 4754
IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA) D. Fu, J. Solinas

RFC 4809
Requirements for an IPsec Certificate Management Profile C. Bonatti, Ed., S. Turner, Ed., G. Lebovitz, Ed.

RFC 4835
Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) V. Manral

RFC 4862
IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten, T. Jinmei

RFC 4868
Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec S. Kelly, S. Frankel

RFC 4869
Suite B Cryptographic Suites for IPsec L. Law, J. Solinas

RFC 4941
Privacy Extensions for Stateless Address Autoconfiguration in IPv6 T. Narten, R. Draves, S. Krishnan

RFC 4945
The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX B. Korver

RFC 5014
IPv6 Socket API for Source Address Selection E. Nordmark, S. Chakrabarti, J. Laganier

RFC 5095
Deprecation of Type 0 Routing Headers in IPv6 J. Abley, P. Savola, G. Neville-Neil

RFC 5175
IPv6 Router Advertisement Flags Option B. Haberman, Ed., R. Hinden

RFC 5282
Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol D. Black, D. McGrew

RFC 5996
Internet Key Exchange Protocol Version 2 (IKEv2) C. Kaufman, P. Hoffman, Y. Nir, P. Eronen

RFC 7627
Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension K. Bhargavan, A. Delignat-Lavaud, A. Pironti, Inria Paris-Rocquencourt, A. Langley, M. Ray

RFC 8446
The Transport Layer Security (TLS) Protocol Version 1.3 E. Rescorla

Internet drafts

Internet drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Other groups can also distribute working documents as Internet drafts. You can see Internet drafts at http://www.ietf.org/ID.html.

Appendix C. Accessibility

Accessible publications for this product are offered through IBM Documentation for z/OS.

If you experience difficulty with the accessibility of any z/OS documentation see How to Send Feedback to IBM to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for the IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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:

IBM Corporation Site Counsel 2455 South Road Poughkeepsie, NY 12601-5400 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, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

Bibliography

This bibliography contains descriptions of the documents in the z/OS Communications Server library.

z/OS Communications Server documentation is available online at the z/OS Internet Library web page at http://www.ibm.com/systems/z/os/zos/library/bkserv/.

z/OS Communications Server library updates

Updates to documents are also available on RETAIN and in information APARs (info APARs). Go to https://www.ibm.com/mysupport to view information APARs.

- z/OS Communications Server V2R1 New Function APAR Summary
- z/OS Communications Server V2R2 New Function APAR Summary
- z/OS Communications Server V2R3 New Function APAR Summary
- z/OS Communications Server V2R4 New Function APAR Summary

z/OS Communications Server information

z/OS Communications Server product information is grouped by task in the following tables.

Planning

Title	Number	Description
z/OS Communications Server: New Function Summary	GC27-3664	This document is intended to help you plan for new IP or SNA functions, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC27-3663	This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server's support of IPv6, coexistence with IPv4, and migration issues.

Resource definition, configuration, and tuning

Title	Number	Description
z/OS Communications Server: IP Configuration Guide	SC27-3650	This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document with the z/OS Communications Server: IP Configuration Reference.

Title	Number	Description
z/OS Communications Server: IP Configuration Reference	SC27-3651	This document presents information for people who want to administer and maintain IP. Use this document with the z/OS Communications Server: IP Configuration Guide. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC27-3672	This document presents the major concepts involved in implementing an SNA network. Use this document with the z/OS Communications Server: SNA Resource Definition Reference.
z/OS Communications Server: SNA Resource Definition Reference	SC27-3675	This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document with the z/OS Communications Server: SNA Network Implementation Guide.
z/OS Communications Server: SNA Resource Definition Samples	SC27-3676	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: IP Network Print Facility	SC27-3658	This document is for systems programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Operation

Title	Number	Description
z/OS Communications Server: IP User's Guide and Commands	SC27-3662	This document describes how to use TCP/IP applications. It contains requests with which a user can log on to a remote host using Telnet, transfer data sets using FTP, send electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC27-3661	This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
z/OS Communications Server: SNA Operation	SC27-3673	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SC27-3665	This document contains essential information about SNA and IP commands.

Customization

Title	Number	Description
z/OS Communications Server: SNA Customization	SC27-3666	This document enables you to customize SNA, and includes the following information: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines

Writing application programs

Title	Number	Description
z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference	SC27-3660	This document 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 document to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC27-3649	This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS® using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC27-3653	This document is for programmers who want application programs that use the IMS TCP/IP application development services provided by the TCP/IP Services of IBM.
z/OS Communications Server: IP Programmer's Guide and Reference	SC27-3659	This document 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. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC27-3674	This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC27-3669	This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC27-3670	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.

Title	Number	Description
z/OS Communications Server: CSM Guide	SC27-3647	This document describes how applications use the communications storage manager.

Diagnosis

Title	Number	Description
z/OS Communications Server: IP Diagnosis Guide	GC27-3652	This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: ACF/TAP Trace Analysis Handbook	GC27-3645	This document explains how to gather the trace data that is collected and stored in the host processor. It also explains how to use the Advanced Communications Function/Trace Analysis Program (ACF/TAP) service aid to produce reports for analyzing the trace data information.
z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures and z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT	GC27-3667 GC27-3668	These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
z/OS Communications Server: SNA Data Areas Volume 1 and z/OS Communications Server: SNA Data Areas Volume 2	GC31-6852 GC31-6853	These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and codes

Title	Number	Description
z/OS Communications Server: SNA Messages	SC27-3671	This document describes the ELM, IKT, IST, IUT, IVT, and USS messages. Other information in this document includes: • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
z/OS Communications Server: IP Messages Volume 1 (EZA)	SC27-3654	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)	SC27-3655	This volume contains TCP/IP messages beginning with EZB or EZD.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC27-3656	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)	SC27-3657	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC27-3648	This document describes codes and other information that appear in z/OS Communications Server messages.

Product Number: 5655-ZOS

SC27-3654-60