z/OS Communications Server

IP Messages: Volume 1 (EZA)

Version 1 Release 4

SC31-8783-02

z/OS Communications Server

IP Messages: Volume 1 (EZA)

Version 1 Release 4

SC31-8783-02

Note:
Before using this information and the product it supports, be sure to read the general information under “Notices” on page 603.

Third Edition (September 2002)

This edition applies to Version 1 Release 4 of z/OS (5694-A01) and Version 1 Release 4 of z/OS.e (5655-G52) and to all subsequent releases and modifications until otherwise indicated in new editions.

Publications are not stocked at the address given below. If you want more IBM® publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Software Reengineering
Department G7IA/ Bldg 503
Research Triangle Park, NC 27709-9990
U.S.A.

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

Fax (USA and Canada):
1-800-254-0206

Internet e-mail:
usib2hpd@vnet.ibm.com

World Wide Web:
http://www.ibm.com/servers/eserver/zseries/zos/webqs.html

IBMLink™:
CIBMORCF at RALVM17

IBM Mail Exchange:
tkinlaw@us.ibm.com

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

Contents

About this document

This document provides information about 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, refer to 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 Reference*. For information about how to use the applications on your TCP/IP system, refer to *z/OS Communications Server: IP User's Guide and Commands*.

This document supports z/OS.e™ .

Who should use 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.

Terms and abbreviations used in this document

This section describes various terms that are used throughout all the IP Messages documents.

How the term "internet" is used in this document

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

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

How the document distinguishes "TCPIP" from "TCP/IP"

The abbreviation TCPIP is used to refer to the specific address space on which the Transmission Control Protocol/Internet Protocol product resides (for example, ″TCPIP continues...″). The abbreviation TCP/IP refers to the product itself.

Other abbreviations and terms

The following abbreviations are also used in this document:

NCS Network Computing System, which is the Apollo implementation of remote procedure calls.

RPC Sun Microsystems' implementation of remote procedure calls.

SQL IBM Structured Query Language.

SQL/DS
IBM Structured Query Language/Data Systems Version 2 Release 2 or later.

ES/9000®
Enterprise System/9000 processor.

ES/9370™
Enterprise System/9370 processor.

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

Term Description

Data set
The basic unit of data storage for MVS™. Unless otherwise specified, the use of this term indicates that the MVS host storing your local data set is your local host system.

Local host
In an internet, any computer to which an end user or a functional unit is connected without the use of the internet.

Remote host
In an internet, any host on the network that requires a physical link to interconnect with the network.

User, local user
Either servers or clients (address spaces) on the local MVS system.

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

The IBM TCP/IP message standards

This section describes the message numbering conventions used in the IP Message manuals.

Older numbering convention

The following diagram shows the older message numbering convention that was used for TCP/IP applications.

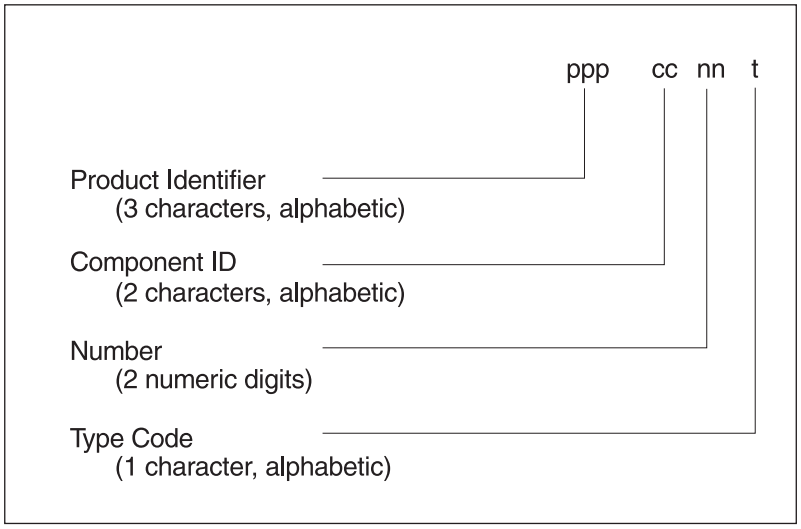


Figure 1. Sample message formats

The various type code characters are defined below.

Type Code
Severity

E or S A recoverable error occurred. Follow the instructions given in the document to fix the problem.

F,T, or U
An unrecoverable error that terminated the program occurred. You might need to contact the IBM Software Support Center.

I Informational message.

N Nondisplay. The message text, but not the message ID, appeared.

W Warning. A condition that might subsequently cause an error has occurred.

X Concatenation. This message is adjoined to one or more messages.

The component identifiers used by some messages with the EZY prefix are listed in the following table.

Table 1. EZY message component identifiers

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

Table 1. EZY message component identifiers (continued)

EZY component ID	Component name
XM	OSF/Motif
XN	OSF/Motif
XO	OSF/Motif
XP	OSF/Motif
XQ	OSF/Motif
XR	OSF/Motif
XU	OSF/Motif
XW	X Window System

Current numbering convention

The following diagram shows the message numbering convention currently in use.

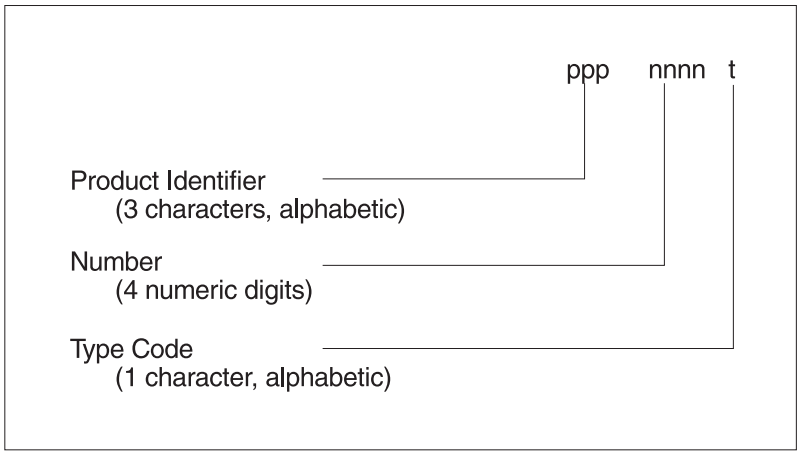


Figure 2. Sample message formats

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

For example:

EZZ0902I Use a host name after HOST option.

The various type code characters are defined below.

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

Message ranges by volume

The IP Messages are distributed among volumes as follows:

Volume 1
EZA Messages

Volume 2
EZB Messages

Volume 3
EZY Messages

Volume 4
EZZ and SNM Messages

Where to find more information

This section contains:

- Pointers to information available on the Internet
- Information about licensed documentation
- Information about LookAt, the online message tool
- A set of tables that describes the documents in the z/OS Communications Server (z/OS CS) library, along with related publications

Where to find related information on the Internet

z/OS

– http://www.ibm.com/servers/eserver/zseries/zos/

z/OS Internet Library

– http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

IBM Communications Server product

– http://www.software.ibm.com/network/commserver/

IBM Communications Server product support

– http://www.software.ibm.com/network/commserver/support/

IBM Systems Center publications

– http://www.redbooks.ibm.com/

IBM Systems Center flashes

– http://www-1.ibm.com/support/techdocs/atsmastr.nsf

RFCs

– http://www.ietf.org/rfc.html

RFC drafts

– http://www.ietf.org/ID.html

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

DNS web sites

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

USENET news groups:
comp.protocols.dns.bind

For BIND mailing lists, see:

- http://www.isc.org/ml-archives/

- 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 (Note: This list may 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.

For definitions of the terms and abbreviations used in this document, you can view or download the latest *IBM Glossary of Computing Terms* at the following Web address:

http://www.ibm.com/ibm/terminology

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

Accessing z/OS licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

`http://www.ibm.com/servers/resourcelink`

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.

To obtain your IBM Resource Link user ID and password, log on to:

`http://www.ibm.com/servers/resourcelink`

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

`http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/`

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

How to contact IBM service

For immediate assistance, visit this Web site:
http://www.software.ibm.com/network/commserver/support/

Most problems can be resolved at this Web site, where you can submit questions and problem reports electronically, as well as access a variety of diagnosis information.

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

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

If you would like to provide feedback on this publication, see “Communicating Your Comments to IBM” on page 609.

z/OS Communications Server information

This section contains descriptions of the documents in the z/OS Communications Server library.

z/OS Communications Server publications are available:

- Online at the z/OS Internet Library web page at
  http://www.ibm.com/servers/eserver/zseries/zos/bkserv
- In softcopy on CD-ROM collections.

Softcopy information

Softcopy publications are available in the following collections:

Titles	Order Number	Description
z/OS V1R4 Collection	SK3T-4269	This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R4, in both BookManager® and PDF formats.
z/OS Software Products Collection	SK3T-4270	This CD includes, in both BookManager and PDF formats, the libraries of z/OS software products that run on z/OS but are not elements and features, as well as the *Getting Started with Parallel Sysplex®* bookshelf.

Titles	Order Number	Description
z/OS V1R4 and Software Products DVD Collection	SK3T-4271	This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager and PDF format. This collection combines SK3T-4269 and SK3T-4270.
z/OS Licensed Product Library	SK3T-4307	This CD includes the licensed documents in both BookManager and PDF format.
System Center Publication IBM S/390® Redbooks™ Collection	SK2T-2177	This collection contains over 300 ITSO redbooks that apply to the S/390 platform and to host networking arranged into subject bookshelves.

z/OS Communications Server library

z/OS V1R4 Communications Server documents are available on the CD-ROM accompanying z/OS (SK3T-4269 or SK3T-4307). Unlicensed documents can be viewed at the z/OS Internet library site.

Updates to documents are available on RETAIN® and in information APARs (info APARs). See Appendix D, "Information APARs" on page 597 for a list of the documents and the info APARs associated with them.

- Info APARs for OS/390® documents are in the document called *OS/390 DOC APAR and PTF ++HOLD Documentation* which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IDDOCMST/CCONTENTS.
- Info APARs for z/OS documents are in the document called *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation* which can be found at http://publibz.boulder.ibm.com:80/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS.

Planning and migration:

Title	Number	Description
z/OS Communications Server: SNA Migration	GC31-8774	This document is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This document also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.
z/OS Communications Server: IP Migration	GC31-8773	This document is intended to help you plan for TCP/IP Services, whether you are migrating from a previous version or installing IP for the first time. This document also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with TCP/IP Services.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC31-8885	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	SC31-8775	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 in conjunction with the *z/OS Communications Server: IP Configuration Reference*.
z/OS Communications Server: IP Configuration Reference	SC31-8776	This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Guide*. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • SMF records • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC31-8777	This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the *z/OS Communications Server: SNA Resource Definition Reference*.
z/OS Communications Server: SNA Resource Definition Reference	SC31-8778	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 in conjunction with the *z/OS Communications Server: SNA Network Implementation Guide*.
z/OS Communications Server: SNA Resource Definition Samples	SC31-8836	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: AnyNet SNA over TCP/IP	SC31-8832	This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.
z/OS Communications Server: AnyNet Sockets over SNA	SC31-8831	This guide provides information to help you install, configure, use, and diagnose sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.
z/OS Communications Server: IP Network Print Facility	SC31-8833	This document is for system 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	SC31-8780	This document describes how to use TCP/IP applications. It contains requests that allow a user to log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC31-8781	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.

Title	Number	Description
z/OS Communications Server: SNA Operation	SC31-8779	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SX75-0124	This document contains essential information about SNA and IP commands.

Customization:

Title	Number	Description
z/OS Communications Server: SNA Customization	LY43-0092	This document enables you to customize SNA, and includes the following: • 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 Application Programming Interface Guide	SC31-8788	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	SC31-8807	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	SC31-8830	This document is for programmers who want application programs that use the IMS™ TCP/IP application development services provided by IBM's TCP/IP Services.
z/OS Communications Server: IP Programmer's Reference	SC31-8787	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	SC31-8829	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	SC31-8811	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.)

Title	Number	Description
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC31-8810	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.
z/OS Communications Server: CSM Guide	SC31-8808	This document describes how applications use the communications storage manager.
z/OS Communications Server: CMIP Services and Topology Agent Guide	SC31-8828	This document describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The document provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis:

Title	Number	Description
z/OS Communications Server: IP Diagnosis	GC31-8782	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: SNA Diagnosis Vol 1, Techniques and Procedures and *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT*	LY43-0088 LY43-0089	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*	LY43-0090 LY43-0091	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	SC31-8790	This document describes the ELM, IKT, IST, ISU, 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)	SC31-8783	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB)	SC31-8784	This volume contains TCP/IP messages beginning with EZB.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC31-8785	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ-SNM)	SC31-8786	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC31-8791	This document describes codes and other information that appear in z/OS Communications Server messages.

APPC Application Suite:

Title	Number	Description
z/OS Communications Server: APPC Application Suite User's Guide	SC31-8809	This documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC application suite. Although its primary audience is the end user, administrators and application programmers may also find it useful.
z/OS Communications Server: APPC Application Suite Administration	SC31-8835	This document contains the information that administrators need to configure the APPC application suite and to manage the APING, ANAME, AFTP, and A3270 servers.
z/OS Communications Server: APPC Application Suite Programming	SC31-8834	This document provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Redbooks

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

Title	Number
TCP/IP Tutorial and Technical Overview	GG24–3376
SNA and TCP/IP Integration	SG24–5291
IBM Communications Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 1: Configuration and Routing	SG24–5227
IBM Communications Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 2: UNIX Applications	SG24–5228
IBM Communications Server for OS/390 V2R7 TCP/IP Implementation Guide: Volume 3: MVS Applications	SG24–5229
Secureway Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements	SG24–5631
TCP/IP in a Sysplex	SG24–5235
Managing OS/390 TCP/IP with SNMP	SG24–5866
Security in OS/390–based TCP/IP Networks	SG24–5383
IP Network Design Guide	SG24–2580
Migrating Subarea Networks to an IP Infrastructure	SG24–5957
IBM Communication Controller Migration Guide	SG24–6298

Related information

For information about z/OS products, refer to *z/OS Information Roadmap* (SA22-7500). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, as well as describing each z/OS publication.

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 table below lists documents that may be helpful to readers.

Title	Number
z/OS Security Server Firewall Technologies	SC24-5922
S/390: OSA-Express Customer's Guide and Reference	SA22-7403

Title	Number
z/OS JES2 Initialization and Tuning Guide	SA22-7532
z/OS MVS Diagnosis: Procedures	GA22-7587
z/OS MVS Diagnosis: Reference	GA22-7588
z/OS MVS Diagnosis: Tools and Service Aids	GA22-7589
z/OS Security Server LDAP Client Programming	SC24-5924
z/OS Security Server LDAP Server Administration and Use	SC24-5923
Understanding LDAP	SG24-4986
z/OS UNIX System Services Programming: Assembler Callable Services Reference	SA22-7803
z/OS UNIX System Services Command Reference	SA22-7802
z/OS UNIX System Services User's Guide	SA22-7801
z/OS UNIX System Services Planning	GA22-7800
z/OS MVS Using the Subsystem Interface	SA22-7642
z/OS C/C++ Run-Time Library Reference	SA22-7821
z/OS Program Directory	GI10-0670
DNS and BIND, Fourth Edition, O'Reilly and Associates, 2001	ISBN 0-596-00158-4
Routing in the Internet , Christian Huitema (Prentice Hall PTR, 1995)	ISBN 0-13-132192-7
sendmail, Bryan Costales and Eric Allman, O'Reilly and Associates, 1997	ISBN 156592–222–0
TCP/IP Tutorial and Technical Overview	GG24-3376
TCP/IP Illustrated, Volume I: The Protocols, W. Richard Stevens, Addison-Wesley Publishing, 1994	ISBN 0-201-63346-9
TCP/IP Illustrated, Volume II: The Implementation, Gary R. Wright and W. Richard Stevens, Addison-Wesley Publishing, 1995	ISBN 0-201-63354-X
TCP/IP Illustrated, Volume III, W. Richard Stevens, Addison-Wesley Publishing, 1995	ISBN 0-201-63495-3
z/OS System Secure Sockets Layer Programming	SC24-5901

Determining if a publication is current

As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager softcopy are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. The following information describes how to determine if you are looking at the most current copy of a publication:

- At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.
- If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.
- To compare softcopy publications, you can check the last two characters of the publication's filename (also called the book name). The higher the number, the

more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

Summary of changes

Summary of changes
for SC31-8783-02
z/OS Version 1 Release 4

This document contains information previously presented in SC31-8783-01, which supports z/OS Version 1 Release 2. 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.

New information

- EZA2294I
- EZA2294I
- EZA2515I–EZA2517I
- EZA2527W–EZA2529I
- EZA2537E
- EZA2601W
- EZA2624W
- EZA2865I–EZA2866I
- EZA2943I–EZA2944I
- EZA2946I
- EZA2947I
- EZA5553I–EZA5560I
- EZA5564I–EZA5571E

An appendix with z/OS product accessibility information has been added.

Changed information:

- EZA2523E–EZA2526E

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Starting with z/OS V1R4, you may notice changes in the style and structure of some content in this document (for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format). The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Also starting with z/OS V1R4, you will notice this book is reorganized by message number to improve retrievability.

This document supports z/OS.e.

Summary of changes
for SC31-8783-01
z/OS Version 1 Release 2

This document contains information previously presented in SC31-8783-00, which supports z/OS Version 1 Release 1.

New information

- EZA1473
- EZA1691 – EZA1696
- EZA1698 – EZA1701
- EZA2104
- EZA2140
- EZA2224 – EZA2228
- EZA2230 – EZA2233
- EZA2423 and EZA2424
- EZA2427 and EZA2428
- EZA2844 and EZA2845
- EZA2850 – EZA2855
- EZA2861 and EZA2862
- EZA2971
- EZA5549 – EZA5552
- EZA5940
- EZA5954

Deleted information

- EZA1470E
- EZA1482I – EZA1484I
- EZA1487I
- EZA1497I
- EZA1499I
- EZA1500I – EZA1506I
- EZA1508I – EZA1510I
- EZA1516I and EZA1517I
- EZA1521I
- EZA1523I – EZA1526I
- EZA1528I – EZA1530E
- EZA1534I
- EZA1537I – EZA1541I
- EZA1646I
- EZA1667I and EZA1668I
- EZA1674W – EZA1679W
- EZA1738W
- EZA1763W
- EZA2088I and EZA2089I
- EZA2246I
- EZA2248I
- EZA5925I
- EZA5949W
- EZA6191W
- EZA9216I and EZA9217I
- EZA9233I

- EZA9452I

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes
for SC31-8783-00
z/OS Version 1 Release 1

This document contains information also presented in *OS/390 V2R10 IBM Communications Server: IP Messages Volume 1 (EZA)*.

Chapter 1. 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.

User or 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.

Source Data Set: 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.

User or 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. Refer to *z/OS Communications Server: IP Configuration Reference* for more information.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified on the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified on the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified on the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified on the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified on the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified in the indicated line of the HOSTS.LOCAL data set and continue. Refer to *z/OS Communications Server: IP Configuration Reference* for more information.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified on the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error identified in the indicated line and continue.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See EZA0530E for more information.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: If this condition is unacceptable, correct the error in the HOSTS.LOCAL data set before continuing.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Reenter the host name in the HOSTNAME parameter of the HOSTS.LOCAL data set and restart the process. Refer to *z/OS Communications Server: IP Configuration Reference* for more information.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: See message number EZA0535W for information.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMMAKSI

Procedure Name: MainProg;

EZA0543I Computing tables ...

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

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: To correct the problem, edit the HOST.LOCAL data set. Discard any duplicate names and reissue MAKESITE.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 refer to *z/OS Communications Server: IP Configuration Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMMASKI

Procedure Name: DumpStatistics

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMMASKI

Procedure Name: DumpStatistics

EZA0567I MVS TCP/IP Makesite

Explanation: This message identifies the operating system and the application.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 refer to *z/OS Communications Server: IP Configuration Reference* .

System Programmer Response: None.

Source Data Set: CMMASKI

Procedure Name: ParmExtract

EZA0578E Output data set file allocation ERROR: *dataset*

Explanation: The HOSTS.SITEINFO data set allocates a limited amount of space for data to be created within the file. To create the output file the data must be entered on a single line up to a maximum of 512 characters per line. The buffer size must be increased to accommodate the data set.

System Action: TCPIP continues.

User or Operator Response: Notify the system programmer to correct the error.

System Programmer Response: By increasing the buffer space, the output data can be stored in the HOSTS.SITEINFO data set.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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, refer to *z/OS Communications Server: IP Configuration Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the gateway, host, or net name to be verified.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If you need to quit, enter QUIT and press ENTER.

System Programmer Response: None.

Source Data Set: CMTSITE

Procedure Name: TestSiteInfo

EZA0601E Error obtaining Userid.

Explanation: The userid entered was not found.

System Action: TCPIP is unsuccessful.

User or Operator Response: Reissue the userid, make sure it is correct.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP halts.

User or Operator Response: Determine the nature of the error, correct it, and try again.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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. Refer to *z/OS Communications Server: IP Configuration Guide* for more information on TCPIP.DATA data set.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 HOSTNAME 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: HOMETEST

Procedure Name: Test

EZA0619I Running IBM MVS TCP/IP *V3R1* 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or Operator Response: Correct the error in the configuration data set and rerun the program. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Turn on ECMODE and reissue the HOMETEST command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Determine why HOMETEST cannot communicate with the name server, and rerun HOMETEST.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP is unsuccessful.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

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

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

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

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: For more information about the error, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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 dataand 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: For more information, contact your 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.

Source Data Set: 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.

User or Operator Response: Resubmit the CONVXLAT command specifying a sequential data set for the output file.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: For more information about the error, contact your 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.

Source Data Set: CMCONVXL PASCAL

Chapter 2. 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: EZAFTPNX

EZA1456I Connect to ?

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

System Action: Processing continues.

User or Operator Response: Enter host name or IP address.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

EZA1459I NAME *host user*

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

System Action: None.

User or 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: None.

Source Data Set: EZAFTPCZ

EZA1460I Command:

Explanation: The system is prompting you to enter a command.

System Action: The system waits for you to enter a command.

User or Operator Response: Enter the command you want to process.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Specify the data set name of the translate table you want to take precedence.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Specify an address space name on the TCP parameter of the FTP command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the error with the data set.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter a valid command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter a correct value using LOCSITE LRECL=value. Refer to 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. Refer to 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.

Source Data Set: EZAFTPCG

Procedure Name: RECVDATA()

EZA1475I Connection with *foreignhost* terminated

Explanation: The connection to the foreign host is closed.

System Action: FTP Client halts.

User or Operator Response: Reestablish a connection with the foreign host's FTP server by using the OPEN subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPSC

EZA1478I GetReply returns *replycode*

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

System Action: FTP continues.

User or 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: None.

EZA1480I Invalid argument string

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

System Action: FTP continues.

User or Operator Response: Correct the argument string by removing the single quotation mark from within the string.

System Programmer Response: None.

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

User or Operator Response: Correct the command string by adding a single quotation mark at the end of the command string.

System Programmer Response: None.

Source Data Set: EZAFTPCP, EZAFTPCG

EZA1485I *number* **bytes transferred.**

Explanation: This message indicates the total number of bytes of binary data transferred to or from the other host.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Increase REGION below the line to 2M.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 64KB. 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.

User or Operator Response: Make sure the appropriate end of record character is in the data to be transmitted.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSC

EZA1542I Usage: *usage*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter more characters of the command and resubmit the request.

System Programmer Response: None.

Source Data Set: EZAFTPPC

EZA1544I TSO *Command*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: Quit or close the current FTP session before attempting to begin a new session.

System Programmer Response: None.

Source Data Set: 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.

User or 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. Refer to the *z/OS Communications Server: IP Configuration Guide* for information on domain name systems or HOSTS.SITEINFO and HOSTS.ADDRINFO data sets.

Source Data Set: 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.

User or 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: None.

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.

User or Operator Response: Specify a port name with a maximum length of 16 bits.

System Action: Open command processing terminates.

System Programmer Response: None.

Source Data Set: EZAFTPCZ

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

Explanation: FTP client is attempting to connect to specified host system.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSC

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

EZA1567I Unknown command: *name*

Explanation: FTP does not recognize the command specified.

System Action: FTP continues.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Specify the indicated subcommand.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Specify the indicated HELP command for syntax information about the subcommand you issued.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

EZA1574I Usage : *value*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA1575I *purpose*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA1577I *command*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA1578I Specify a command by any unambiguous prefix

Source Data Set: EZAFTPCL

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

Source Data Set: EZAFTPCL

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

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA1583I Note: no password was requested

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

EZA1585I leaving dologin

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

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.

User or 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: None.

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 HFS the delimiter is a forward slash (/), otherwise the delimiter will be a period (.).

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: Issue the DEBUG subcommand to toggle on the debug option.

System Programmer Response: None.

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.

User or Operator Response: If the data transfer was not successful, reestablish the control connection and attempt the data transfer again.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the FTP command to reestablish the data connection.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCG

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.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA1681W Local file already exists

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Correct the file identifier and resubmit the command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Resubmit the command with the correct file identifier.

System Programmer Response: None.

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.

User or 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: None.

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.

User or Operator Response: Check to make sure the data set exists. If so, resubmit the command with the correct data set name.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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 filetype is not valid for **srestart**. Only filetype=SEQ is valid for **srestart**.

type is the current filetype setting.

System Action: FTP stops processing **srestart** and prompts for another subcommand.

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

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Use the **get** or **put** subcommands instead of **srestart** to transfer the file.

System Programmer Response: None.

Source Data Set: 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

User or 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: None.

Source Data Set: 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.

User or 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*

Source Data Set: 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.

User or Operator Response: Report the error to the system programmer.

System Programmer Response: Verify that the source and target 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

Procedure Name: dohelp()

EZZ1701I >>>

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

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.

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

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

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.

User or 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.

EZA1735I FTP 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 return code. *rc* is in the format **yyxxx**, where

yy the subcommand code, which is a number from 0 to 99. See the *z/OS Communications Server: IP User's Guide and Commands* for a list of the subcommand codes.

xxx This is the number of the FTPD Reply Code that immediately precedes this message. See the *z/OS Communications Server: IP and SNA Codes* for the list of FTPD Reply Codes.

ec is the internal CLIENT ERROR CODE set by the FTP CLIENT. For more information about these codes, see the *z/OS Communications Server: IP User's Guide and Commands*.

System Action: FTP ends. TCPIP continues.

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

System Programmer Response: See the preceding FTPD Reply Code and the *z/OS Communications Server: IP User's Guide and Commands* for information about FTP Subcommand Codes and FTP Client Error Codes to determine the cause of the error.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCU

Source Data Set: EZAFTPCX

EZA1737E Command input too long.

Explanation: The FTP command input line exceeds the expected length.

System Action: FTP bypasses the command and continues.

User or Operator Response: Correct the input command stream.

System Programmer Response: None.

Source Data Set: EZAFTPCU

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.

User or Operator Response: Correct the value specified for the EXIT= start parameter.

System Programmer Response: None.

Source Data Set: EZAFTPCY

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCY

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCY

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 target 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.

User or 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: None.

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.

User or Operator Response: Contact the administrator for the file and request access.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA1788I ACCOUNT

Explanation: This is a prompt for accounting information from the target host system.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Consult your 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.

Source Data Set: EZAFTPCT

Chapter 3. 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.

User or Operator Response: None.

System Programmer Response: None.

EZA2064I Local directory is *directory*

Explanation: The current local directory.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA2065I Local directory is null string.

Explanation: No local directory name has been specified.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCD

EZA2071I Mismatched quotes on directory name *"name"*

Explanation: The syntax used in the LCD parameter is incorrect.

System Action: FTP continues.

User or Operator Response: Enter the command 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: None.

Source Data Set: 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.

User or 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: None.

EZA2073I A qualifier in *pathname* begins with an invalid character.

Explanation: The path name specified in the lmkdir command is incorrect.

System Action: FTP continues.

User or Operator Response: Enter the command with a valid path name and try it again.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None

System Programmer Response: None

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCD, EZAFTPCL

EZA2082I Local directory name set to *directory*

Explanation: This message displays the name of the local directory.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCB

Procedure Name: srestart_get(), srestart_put()

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 are in blocks. Depending on the values assigned to these parameters, only one of these messages will be displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 are in cylinders. Depending on the values assigned to these parameters, only one of these messages will be displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 are in tracks. Depending on the values assigned to these parameters, only one of these messages will be displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2152I Volid of Migrated Data Sets is *id*

Explanation: The volume ID for allocated new migrated data sets.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2153I Wrapping data into next record.

Explanation: Data is being sent as part of the next record.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2187I Free Free Largest Free

EZA2188I Volume Unit Free Cyls Trks Cyls-Trks Exts Address Use Attr

Explanation: These messages display the headers for the information provided when the Qdisk parameter has been specified for the LOCSITE subcommand. This message is displayed with EZA2191.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2191I *volume unit amount value value value value value vary address attr1 attr2*

Explanation: The Qdisk parameter has been specified for the LOCSITE subcommand. This parameter provides statistics about the available space on the specified volume. If Qdisk is entered without a specific volume serial, statistics about available space on all storage volumes are displayed. This message is displayed with EZA2187I and EZA2188I.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: Resubmit the SITE command with a valid value for the SQLCOL parameter. For more information about the SITE command, see the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 C/C++ Programming Guide*.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

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

System Programmer Response: Allocate additional storage for the address space.

Source Data Set: 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 is an excerpt 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 C, “Related protocol specifications (RFCs)” on page 587 for information about accessing RFCs.

System Action: FTP stops trying to connect to the indicated IP address through the SOCKS server.

User or 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.

Source Data Set: 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 C, “Related protocol specifications (RFCs)” on page 587 for information about accessing RFCs.

System Action: The FTP client stops trying to connect to the FTP server through the SOCKS server.

User or 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.

Source Data Set: 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 is an excerpt 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 C, "Related protocol specifications (RFCs)" on page 587 for information about accessing RFCs.

System Action: FTP stops trying to access the FTP server through the SOCKS server.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

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*.

User or Operator Response: If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECfm, and BLOCKSIze combination. Refer to 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: None.

Source Data Set: 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*.

User or Operator Response: If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. Refer to 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: None.

Source Data Set: 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*.

User or Operator Response: If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. Refer to 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: None.

Source Data Set: 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*.

User or Operator Response: If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. Refer to 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: None.

Source Data Set: 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*.

User or Operator Response: If necessary, reissue the LOCSITE subcommand with a valid LRecl, RECFm, and BLOCKSIze combination. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: locstat()

Procedure Name: ezaftpcl

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2321I Error recalling *temp* **(rc=value)**

Explanation: The file displayed cannot be recalled.

System Action: FTP continues.

User or Operator Response: Check the AUTORECALL/NOAUTORECALL parameter in the *hlq*.FTP.DATA data set. If you would like automatic recall of data sets, set the parameter to AUTORECALL. For more information on the AUTORECALL/NOAUTORECALL parameter, see the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2322I *temp* **is on a direct access volume that is not mounted and Noautomount is specified.**

Explanation: The default value for the FTP AUTOMOUNT parameter is set to NOAUTOMOUNT, preventing automatic mounting of volumes. The file displayed is on a volume that has not been mounted.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2323I Error mounting volume for *temp*

Explanation: FTP is unable to mount the volume for the file displayed.

System Action: FTP continues.

User or Operator Response: Check the AUTOMOUNT/NOAUTOMOUNT parameter in the *hlq*.FTP.DATA data set. If you would like auto mounting of data sets, set the parameter to AUTOMOUNT. For more information on the AUTOMOUNT/NOAUTOMOUNT parameter, see the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: EZAFTPCD, EZAFTPCG, EZAFTPCP

EZA2326I Error allocating data set *name*

Explanation: The data set name displayed is not valid.

System Action: FTP continues.

User or 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.

Source Data Set: EZAFTPCD

EZA2329I No files match the pattern *name*

Explanation: There are no files or data sets that match the pattern.

System Action: FTP continues.

User or Operator Response: Retry the command with the correct pattern.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check the *z/OS Communications Server: IP User's Guide and Commands* for information on naming requirements of data sets. Retry the command with the correct name.

System Programmer Response: None.

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.

User or 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: None.

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 path name, 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.

User or 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: None.

EZA2337I *temp* **is not a valid file.**

Explanation: The file displayed is not valid.

System Action: FTP continues.

User or Operator Response: Reissue the command with a valid file name.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2338I *temp* **is not a partitioned data set.**

Explanation: The name displayed is not a partitioned data set.

System Action: FTP continues.

User or 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: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE subcommand with a valid parameter. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If desired, reissue the LOCSITE command with an operand for the *parm* option. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If desired, reissue the LOCSITE command without an operand on the *parm* parameter. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE LRecl command with a record size between 0 and 32760.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE BLOCKSIze command with a block size between 0 and 32760.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCK

EZA2398W *parameter* **is not a valid unit parameter.**

Explanation: The unit parameter is not valid.

System Action: FTP continues.

User or Operator Response: Reissue the unit name command with the correct parameter.

System Programmer Response: FTP continues.

Source Data Set: 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.

User or Operator Response: Issue another locsite subcommand to change the data set filetype. 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCK, EZAFTPEP

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.

User or 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: None.

Source Data Set: EZAFTPCK

EZA2407W Mgmtclass parameter *parameter* **is not valid.**

Explanation: The MGmtclass parameter displayed is not valid.

System Action: FTP continues.

User or 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: None.

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.

User or 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: None.

Source Data Set: EZAFTPCK

EZA2409W STORclass parameter (*parameter*) is not valid.

Explanation: The STORclass parameter displayed is not valid.

System Action: FTP continues.

User or 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: None.

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.

User or 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: None.

Source Data Set: EZAFTPCK

EZA2411W DATAClass parameter *parameter* is not valid.

Explanation: The DATAClass parameter displayed is not valid.

System Action: FTP continues.

User or 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: None.

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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: To discard RDW during FTP transmission, set RDW to false in the *hlq*.FTP.DATA data set.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCK

EZA2415W RETpd parameter ***(value)*** **is not valid.**

Explanation: The RETpd parameter displayed is not valid.

System Action: FTP continues.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Correct the name of the DB2 subsystem, and issued the LOCSITE command again.

System Programmer Response: None

Source Data Set: 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.

User or Operator Response: Change value to one of the supported formats.

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

EZA2421W NORESTGet already specified. RESTGet ignored.

Explanation: NORESTGet cmd was follow by a RESTGet cmd within the same LOCSITE request.

System Action: FTP continues.

User or Operator Response: If the setting is not what you want, reissue the LOCSITE command with the correct values.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2422W RESTGet already specified. NORESTGet ignored.

Explanation: RESTGet cmd was follow by a NORESTGet cmd within the same LOCSITE request.

System Action: FTP continues.

User or Operator Response: If the setting is not what you want, reissue the LOCSITE command with the correct values.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with valid values.

System Programmer Response: None.

Source Data Set: 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 HFS path

Userid The FTP userid

Local The current directory from the LCD command

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

Procedure Name: locstat()

EZA2491I Compressed mode implemented with type EBCDIC only.

Explanation: Compressed file transfers are implemented in EBCDIC format only.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

EZA2494I Checkpoint interval is *interval*

Explanation: The interval for generating checkpoint markers is displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCB

EZA2501W Restart requires EBCDIC data type

Explanation: A RESTART command was issued, but the representation type is not EBCDIC.

System Action: FTP continues.

User or Operator Response: Change the representation type to EBCDIC and reissue the RESTART command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Change the transfer mode to block or compressed and reissue the RESTART command.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2509I *number* **megabytes transferred.**

Explanation: This message indicates the total number of bytes of binary data transferred to or from the other host. Megabyte is defined as 1024*1024 bytes.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reenter the LOCSITE command to change the option.

System Programmer Response: None.

Source Data Set: locsite()

Procedure Name: EZAFTPCK

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.

User or Operator Response: If this is not the intended result, issue the LOCSITE TRUNcate command.

System Programmer Response: None.

Source Data Set: locstat()

Procedure Name: EZAFTPCL

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.

User or Operator Response: If this is not the intended result, issue the locsite TRUNcate command.

System Programmer Response: None.

Source Data Set: locstat()

Procedure Name: EZAFTPCL

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: locstat

Procedure Name: EZAFTPCL

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Either specify the matching READTAPEFormat or change it to unspecified and retry the request. Refer to the *z/OS Communications Server: IP User's Guide and Commands* for the description of the LOCSITE command for quidelines on using READTAPEFormat.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Either specify the matching READTAPEFormat or change it to unspecified and retry the request. Refer to the *z/OS Communications Server: IP User's Guide and Commands* for the description of the LOCSITE command for quidelines on using READTAPEFormat.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Either specify the matching READTAPEFormat or change it to unspecified and retry the request. Refer to the *z/OS Communications Server: IP User's Guide and Commands* for the description of the LOCSITE command for quidelines on using READTAPEFormat.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Either specify the matching READTAPEFormat or change it to unspecified and retry the request. Refer to the *z/OS Communications Server: IP User's Guide and Commands* for the description of the

LOCSITE command for quidelines on using READTAPEFormat.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reenter the LOCSITE command to change the option.

System Programmer Response: None.

Source Data Set: locsite()

Procedure Name: EZAFTPCK

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.

User or Operator Response: If this is not the intended result, issue the LOCSITE NOTRUNcate command.

System Programmer Response: None.

Source Data Set: locstat()

Procedure Name: EZAFTPCL

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 a Gigabyte, message EZA1617I is displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Increase the value of LRECL for the new file or change the LOCSITE options.

System Programmer Response: None.

Source Data Set: rcvfile()

Procedure Name: EZAFTPCG

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.

User or Operator Response: Correct the data set name and reissue the request.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Contact the owner of the data set for authorization to read the data set.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: EZAFTPMR

Source Data Set: 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.

User or Operator Response: Issue the LOCSITE AUTORECALL command to allow the data set to be recalled, and then reissue the FTP command.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Verify that the PDS name was entered correctly.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Verify that the correct data set name was entered. Reissue the command with the correct data set name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer.

System Programmer Response: Correct the problem with the data set.

Source Data Set: EZAFTPMR

EZA2550W Name length error for filename *name*

Explanation: The path name 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 filename is 255 -- the total path name maximum length is 1023.

System Action: The request is not performed.

User or Operator Response: Reissue the request with a name that meets the limits for the type of data set or file.

System Programmer Response: None.

Source Data Set: 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 filename 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.

User or Operator Response: Verify that the OMVS user_id has a HOME directory. If necessary, define a HOME directory for the user_id.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid relative number for the data set.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Ensure that the GDG exists.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command specifying a name of a file that is not a directory.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command specifying a name of a file that is not a pipe or FIFO special file.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command specifying a name of a file that is not a character special file.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command specifying a name of a file that is not an unknown file type.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command specifying a name of a file exists.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *alloc_tape*. These trace messages provide the reason codes for the dynamic allocation error.

Source Data Set: 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.

dsn is the data set name.

s99error is the hexadecimal s99error code from an SVC 99 request or a DYNALLOC request. See the *z/OS MVS Programming: Authorized Assembler Services Guide* for a list of the s99error codes.

s99info is the hexadecimal s99info code from an SVC 99 request. See the *z/OS MVS Programming: Authorized Assembler Services Guide* for a list of the s99info codes.

s99ersn is the hexadecimal s99ersn code from SVC 99 request. See the *z/OS MVS Programming: Authorized Assembler Services Guide* for a list of the s99ersn codes.

System Action: The command is rejected. FTP continues.

User or Operator Response: None.

System Programmer Response: Examine the trace and look for trace messages with the tag *alloc_dasd*. These trace messages provide the reason codes for the dynamic allocation error. Please refer to *z/OS MVS Programming: Authorized Assembler Services Guide* for information regarding the error codes and information codes returned from dynamic allocation services.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

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.

User or Operator Response: Try the operation later when no one else is using the member of the PDS.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2564W Open of *dsname* failed.

Explanation: The data set cannot be opened.

System Action: The requested operation is terminated. FTP continues.

User or Operator Response: None.

System Programmer Response: Look at the trace and find the following trace entry sequence:

```
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 error that occurred. *aa* is the errno value returned from the fopen of file. **bbbb** is the text associated with the errno value. The following describe values and text that could be observed:

(61): Error trying to define file
The file is already opened.

Source Data Set: 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.

User or Operator Response: Contact the owner of the data set for authorization to read the data set.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Look at the trace to determine the error that occurred when the repositioning was done.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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 the *DB2 Messages and Codes* for a complete description of SQL return codes.

System Action: No data is sent. FTP continues.

User or 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 your system programmer for FTP.

System Programmer Response: Ensure that the DB2 subsystem has been started.

Source Data Set: 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 the *DB2 Messages and Codes* for a complete description of SQL return codes.

System Action: No data is sent. FTP continues.

User or Operator Response: If FTP SQL support should be available, contact your system programmer for FTP. If a SQL query was not intended, issue **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.

Source Data Set: 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.

User or Operator Response: Correct the error indicated in the output file, and resubmit the SQL request.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If possible, correct the error indicated in the output file and resubmit the SQL request. Otherwise, contact your system programmer for FTP.

System Programmer Response: Correct the error indicated by the message contained in the output file. For a detailed explanation of DB2 error codes, see the *DB2 Messages and Codes*.

Source Data Set: 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.

User or Operator Response: If possible, correct the error indicated in the output file and resubmit the SQL request. Otherwise, contact your system programmer for FTP.

System Programmer Response: Correct the error indicated by the message contained in the output file. For detailed explanation of DB2 error codes, see the *DB2 Messages and Codes*.

Source Data Set: EZAFTPCP

EZA2573E Transfer aborted: unsupported SQL statement

Explanation: FTP retrieved a file (or data set) while the local filetype was SQL, but the file contains an SQL statement that is not supported by FTP.

System Action: The PUT command is rejected.

User or 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: None.

Source Data Set: 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 the *DB2 Messages and Codes* for a complete description of SQL return codes.

System Action: FTP continues.

User or 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 the *DB2 Messages and Codes* for a complete description of SQL return codes.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

Source Data Set: 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.

User or 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*.

Source Data Set: 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 HFS directory indicated by *directory*.

System Action: The current working directory is changed to the new directory.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 path 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.

User or Operator Response: Verify that the OMVS user_id has a HOME directory. If necessary, define a HOME directory for the user_id.

System Programmer Response: None.

Source Data Set: 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 HFS directory *directory*
- EZA2586I for PUT is "*directory*" name prefix.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 path name of ../ to back up one directory level, but the current working directory was either an MVS partitioned data set, or an MVS high level qualifier. The "../" notation is only valid when the current working directory is an HFS directory.

System Action: The LCD command is rejected.

User or Operator Response: Reissue the LCD command with the corrected path name.

System Programmer Response: None.

Source Data Set: EZAFTPCD

EZA2588E Unable to connect to alternate TCPIP *name*

Explanation: Error returned from setibmopt routine.

System Action: None.

User or 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: None.

Source Data Set: EZAFTPCY

EZA2589E Connection to server interrupted or timed out.

Explanation: The FTP server at the specified host is no longer responding to the FTP client or the user has caused an interrupt (for example, Cntrl-c).

System Action: The current subcommand is terminated.

User or 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.

System Programmer Response: None.

Source Data Set: EZAFTPSC

EZA2590E *name* **error from** *name* **= (***number* **.** *number***) -** *string*

Explanation: The FTP client has issued a socket call and received an error return code.

System Action: Communications are interrupted. Current operation terminated.

User or Operator Response: Retry.

System Programmer Response: None.

Source Data Set: EZAFTPSC

EZA2591I Waiting for recall of "*dsname***"**

Explanation: A LCD command was issued to change the current working directory to *dsname*. *dsname* was an existing MVS data set which was migrated. The client needed to recall the data set in order to determine whether or not the data set was a partitioned data set.

System Action: The client attempts to recall the data set.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCD

EZA2592I Waiting for mount for "*dsname*"

Explanation: An LCD command was issued to change the current working directory to *dsname*. The data set was an existing MVS data set that was catalogued on an unmounted volume. The client needed to mount the volume for the data set in order to determine whether or not the data set was a partitioned data set.

System Action: The client waits for the volume to be mounted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCD

EZA2593I HFS directory "*directory*" created.

Explanation: The indicated directory has been created.

System Action: The directory is created. FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Contact the security administrator to get the necessary authorization for creating the data set.

System Programmer Response: None.

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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

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. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: 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.

User or Operator Response: Reissue the command without the proxy prefix, or enter a valid command for the proxy server.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Issue the LOCSITE subcommand again with the desired setting.

System Programmer Response: None.

EZA2602E Error reading the file

Explanation: The local file could not be read.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCP

EZA2603E Error sending the file

Explanation: The local file could not be sent.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Set the structure to FILE, the mode to STREAM, the transfer type to IMAGE, and reissue the command.

System Programmer Response: None.

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.

User or Operator Response: Request a larger space allocation for the local data set and try the transfer again.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

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

System Programmer Response: Correct the error described in *file_status*.

Source Data Set: 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.

| **User or Operator Response:** If you want to use *parameter1*, issue a LOCSITE subcommand specifying *parameter1*,
| but not *parameter2*. Refer to 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:** None.

| **Source Data Set:** 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.

| **User or Operator Response:** Issue the LOCSITE subcommand again with the desired setting.

| **System Programmer Response:** None.

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.

User or Operator Response: Reissue the LOCSITE command with a valid MVS data set name. Refer to 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with a valid value specified for the CHKptint parameter. Refer to 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: None.

Source Data Set: 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 HFS maximum of 1023 characters.

System Action: The command is rejected.

User or Operator Response: Reissue the command with the correct data set or HFS file name.

System Programmer Response: None.

Source Data Set: EZAFTPMR

EZA2633I File name too long. Use HFS naming conventions.

Explanation: The command was entered with a file name that was longer than the HFS maximum of 1023 characters.

System Action: The command is rejected.

User or Operator Response: Reissue the command with the correct HFS file name.

System Programmer Response: None.

Source Data Set: 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 path name 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.

User or Operator Response: Verify that the OMVS user_id has a HOME directory. If necessary, define a HOME directory for the user_id.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with the correct member name syntax.

System Programmer Response: None.

Source Data Set: EZAFTPMR

EZA2636E TSO *Command* **failed with return code** *value*

Explanation: The TSO command requested completed in error.

System Action: The command fails.

User or 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.

Source Data Set: 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 path name.

System Action: The command is rejected.

User or Operator Response: Verify that the correct data set name or path was entered.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the command. If the problem persists, contact the system programmer.

System Programmer Response: Correct the problem with the data set.

Source Data Set: 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.

User or Operator Response: Correct the data set name and reissue the command.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: EZAFTPMR

EZA2641I Patterns (% and *) may only be specified in the last qualifier when in directory mode

Explanation: The command was entered with a path name that contained the global file-name characters % and *. The global file-name characters were not in the last qualifier of the path name, and the MVS server was currently in directory mode. The global file-name characters can only be used in the last qualifier when in directory mode.

System Action: The command is rejected.

User or Operator Response: Either correct the path name to have the global file-name characters in the last qualifier and reissue the command in directory mode, or issue the LOCSITE/SITE DATASETMODE command to change the server to data set mode and then reissue the command as is.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Retry the command. 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.

Source Data Set: 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.

User or Operator Response: Verify that the member name or path was entered correctly.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reenter the LOCSITE subcommand, specifying the correct value for *parm*. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value. Refer to 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: None.

Source Data Set: EZAFTPCK

EZA2660I Workarea allocation failed.

Explanation: The allocation of the workarea for qdisk failed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2661W Restart requires file structure

Explanation: A RESTART command was issued, but the structure is not file structure.

System Action: FTP continues.

User or Operator Response: Change the structure to file (issue file or structure f) and reissue the RESTART command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None required, the parameter is ignored.

System Programmer Response: None.

Source Data Set: 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 userid. The user must first login with a valid userid and password before issuing the LOCSITE subcommand with the *parameter* parameter.

System Action: FTP continues.

User or Operator Response: Login to the server using the USER and PASS subcommands, then reissue the LOCSITE subcommand with the *parameter* parameter.

System Programmer Response:

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with the correct syntax for the CHmod parameter. Refer to 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: None.

EZA2665W CHMod invalid filename *filename*

Explanation: The *filename* specified by the CHMod parameter of the LOCSITE subcommand was longer than the HFS filename maximum of 1023 characters. If the filename operand of the CHMod parameter began with a forward slash (/), then *filename* will be the filename operand used "as is"; otherwise *filename* will be the filename operand appended to the current working directory.

System Action: The CHMod parameter is ignored.

User or Operator Response: Reissue the LOCSITE command with a valid value specified for the filename on the CHMOD parameter. Refer to 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: None.

Source Data Set: EZAFTPCK

EZA2667W CHMod filename is not a valid HFS file. CHMod ignored

Explanation: The filename specified on the CHMod parameter of the LOCSITE subcommand was not an HFS file. The CHMod parameter is only valid for HFS files.

System Action: The CHMod parameter is ignored.

User or Operator Response: Reissue the LOCSITE command with a valid value specified for the filename on the CHMod parameter. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with a valid value specified for the filename on the CHMod parameter. If the problem persists, contact the system programmer.

System Programmer Response: If necessary, correct the error indicated by *error_description*.

Source Data Set: 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.

User or Operator Response: Contact the system programmer.

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

Source Data Set: 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.

User or Operator Response: Reenter the LOCSITE subcommand, specifying the correct value for the mode operand. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: Reenter the LOCSITE subcommand, specifying the correct operands. Refer to 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with a valid value for the UMask parameter. Refer to 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.

Source Data Set: EZAFTPCK

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.

User or Operator Response: Reissue the LOCSITE command with a valid value for the UMask parameter. Refer to 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.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with a valid value for the UMask parameter. Refer to 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2690I Invalid response to prompt. Try again.

Explanation: You entered an invalid response to an interactive prompt.

System Action: FTP continues.

User or Operator Response: Enter one of the valid repsonses that are displayed in the prompt message.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA2692I Globbing off

Explanation: Filename 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 filename 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 filename 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

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

Source Data Set: EZAFTPNC

EZY2706E Command not available. Unable to access Japanese translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: 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.

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

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

Source Data Set: EZAFTPNC

EZY2708E Command not available. Unable to access Korean translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: 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.

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

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

Source Data Set: EZAFTPNC

EZY2710E Command not available. Unable to access traditional Chinese translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: 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.

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

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

Source Data Set: EZAFTPNC

EZY2712E Command not available. Unable to access simplified Chinese translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command you want to process.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Issue the OPEN subcommand with a host name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Issue the command without replace.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Issue the command without notype, or not as a proxy command.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 catalogued if a store operation terminates abnormally.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 filename.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 HFS files.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 HFS file containing translate tables generated by the CONVXLAT utility.

System Action: FTP continues.

User or Operator Response: Resubmit the LOCSITE subcommand with corrected syntax.

System Programmer Response: None.

Source Data Set: 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 a invalid code page name. The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Refer to the *z/OS C/C++ Programming Guide* for information on supported code set converters and valid code set names. Resubmit the corrected LOCSITE subcommand.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Refer to the *z/OS 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: None.

Source Data Set: 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.

User or Operator Response: Ensure that the requested code set names are for single-byte code pages. SBDATACONN is not supported for double-byte. Refer to the *z/OS C/C++ Programming Guide* for information on supported code set converters and valid code set names.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2738W Translate file '*file*' not found. SBDATACONN ignored.

Explanation: A LOCSITE subcommand has been entered with an SBDATACONN parameter that specified a filename, but FTP was unable open *file*. The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Ensure that the filename specified on the SBDATACONN parameter is a fully qualified name of an existing MVS data set or HFS file.

System Programmer Response: If the filename is correct, use the FTP client's debug facility to determine why FTP cannot open the file.

Source Data Set: EZAFTPCK

EZA2739W Translate file '*file*' invalid. SBDATACONN ignored.

Explanation: A LOCSITE subcommand has been entered with an SBDATACONN parameter that specified a filename, but FTP encountered an error while trying to load its translate tables using *file*. Possible errors include an invalid header record or incorrect file length. (The file must contain exactly 768 bytes of data.) The SBDATACONN parameter is ignored.

System Action: FTP continues.

User or Operator Response: Ensure that the filename specified is correct. It must be a fully qualified name, and the file must contain translate tables in the format generated by the CONVXLAT utility.

System Programmer Response: Use the client's debug facility to help determine why the file cannot be used.

Source Data Set: 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.

User or Operator Response: If FTP SQL support is desired, 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 filetype from its current setting of 'SQL' to 'SEQ'.

System Programmer Response: Ensure that the appropriate DSNLOAD library is accessible.

Source Data Set: 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.

User or Operator Response: Reenter the LOCSITE command with corrected syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command, specifying either ″7bit″ or the name of an ASCII code page supported by iconv. Refer to the *z/OS C/C++ Programming Guide* for a list of the code pages supported by iconv. The code page name must be entered exactly as shown in the list (for example: IBM-850 or ISO8859-1).

System Programmer Response: None

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: If you wish 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: None.

Source Data Set: 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: None.

User or Operator Response: The data set can be deleted with the data set utilities of the operating system.

System Programmer Response: None.

Source Data Set: EZAFTPCG

EZA2746E A qualifier in *name* begins with an invalid character.

Explanation: A data set qualifier specified on the LCD or LMKDIR commands begins with an invalid character.

System Action: None.

User or Operator Response: Reissue the command with a valid data set name.

System Programmer Response: None.

Source Data Set: EZAFTPGU

EZA2747E A qualifier in *name* is more than 8 characters.

Explanation: A data set quailifier specified on the LCD or LMKDIR command is longer than 8 characters.

System Action: None.

User or Operator Response: Reissue the command with a valid data set name.

System Programmer Response: None.

Source Data Set: EZAFTPGU

EZA2748E A qualifier in *name* contains an invalid character.

Explanation: A data set quailifier specified on the LCD or LMKDIR command contains an invalid character.

System Action: None.

User or Operator Response: Reissue the command with a valid data set name.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Correct the FTP.DATA file to contain the correct code set for UCSHOSTCS. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: 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.

User or Operator Response: Reissue the LOCSITE command with a valid codeset name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2773E TYPE U transfer is not supported in mode *mode*

Explanation: FTP has retreived 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.

User or Operator Response: Change the mode to stream using the STREAM command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2775E Record structure is not supported for TYPE U transfer

Explanation: FTP has retreived 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.

User or Operator Response: Change the structure to FILE using the STRUCT=F command.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2776E TYPE U transfer is not supported for filetype *type*

Explanation: FTP has retreived 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 transfered as UCS-2 data.

System Action: The GET or PUT command is rejected.

User or Operator Response: If the file is not JES or SQL, change the filetype to SEQ using the LOCSITE FILETYPE=SEQ command.

System Programmer Response: None.

Source Data Set: EZAFTPCG, EZAFTPCP

EZA2777I TYPE U transfer is only supported for filetype SEQ

Explanation: An UCS-2 transfer failed because the filetype was not sequential. See EZA2776E for more information.

System Action: FTP continues.

User or Operator Response: None

System Programmer Response: None

Source Data Set: 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.

User or Operator Response: Turn RDWs off using the LOCSITE NORDW command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Turn TRAILINGBLANKS off using the LOCSITE NOTRAILINGBLANKS command.

System Programmer Response: None.

Source Data Set: 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 storing UCS-2 data in EBCDIC format into a ASA data set.

System Action: The GET or PUT command is rejected.

User or Operator Response: Turn ASATRANS on using the LOCSITE ASATRANS command.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Turn WRAPRECORD off using the LOCSITE NOWRAPRECORD command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Change the target data set to a variable format data set.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCG

EZA2788I Command ! may only be issued when the client is running in the OpenEdition® shell.

Explanation: This message is issued when the ! command was issued while not under the z/OS UNIX System Services shell.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: To avoid this message, remove the CCTRANS statement from the FTP.DATA file.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: To avoid this message, remove the SBTRANS statement from the FTP.DATA file.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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: None.

User or Operator Response: Try again later.

System Programmer Response: None.

Source Data Set: EZAFTPCP, EZAFTPCG

EZA2800E The FTP client is not available because the TCP/IP Base feature of OS/390 is not enabled.

Explanation: The FTP client attempted to register as product=OS/390 feature=TCP/IP_BASE but IFAPRDxx parmlib member does not indicate that the feature TCP/IP_BASE is enabled. This FTP client will only run if TCP/IP_BASE is enabled.

System Action: The FTP initialization is terminated.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the IFAPRDxx parmlib member if it is in error and restart the FTP client. If TCP/IP_BASE is not part of your system there is no action to take and this FTP client cannot be used.

Source Data Set: EZAFTPCY

EZA2801I Data not wrapped into next record.

Explanation: Data that does not fit within the target record will be truncated.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: See message EZA2805W for the system completion code and reason code and take actions appropriate for the failure.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Refer to the *z/OS MVS System Codes* publication for suggested actions for the specified failure code.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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 specifed 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCY

EZA2808W File has unsupported DSORG.

Explanation: Requested file has an DSORG that is not supported by FTP.

System Action: The file transfer request is terminated. FTP continues.

User or Operator Response: Check DSORG for the file.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA2814I UCOUNT is *ucount*

Explanation: UCOUNT is the unit count for new allocations.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA2815I VCOUNT is *vcount*

Explanation: VCOUNT is the volume count for new allocations.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: EZAFTPCL

Source Data Set: None.

EZA2816I No automatic mount of tape volumes.

Explanation: NOAUTOMOUNT was specified. NOAUTOMOUNT is the default value.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA2817I Automatic mount of tape volumes.

Explanation: AUTOMOUNT was specified.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

EZA2822W NOFWFriendly already specified. FWFriendly ignored.

Explanation: NOFWFriendly was already specified on this LOCSITE command, FWFriendly is ignored.

System Action: FTP continues.

User or Operator Response: Specify one option, FWFriendly or NOFWFriendly on the LOCSITE command.

System Programmer Response: None.

Source Data Set: EZAFTPCK

EZA2823W FWFriendly already specified. NOFWFriendly ignored.

Explanation: FWFriendly was already specified on this LOCSITE command, NOFWFriendly is ignored.

System Action: FTP continues.

User or Operator Response: Specify one option, FWFriendly or NOFWFriendly on the LOCSITE command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Override the incorrect BLKSIZE/LRECL by issuing a SITE command with the correct settings.

System Programmer Response: None.

Source Data Set: EZAFTPCP

Procedure Name: sndfile

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.

User or 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: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Reissue the command with a valid value.

System Programmer Response: None.

Source Data Set: 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 e-mail address.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Correct the FTP.DATA file to contain the correct value for the specified parameter. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: 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.

User or Operator Response: If you want load module transfer processing to be performed, issue a LOCSITE FILETYPE=SEQ subcommand.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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 HFS 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.

User or Operator Response: If you want load module transfer processing to be performed, reissue the transfer subcommand without using quotes or slashes.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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 filename provided.

System Action: FTP waits for the next subcommand.

User or Operator Response: Check the filename specified on the PUT or MPUT subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCC

EZA2837I Could not build a file list

Explanation: While performing load module transfer, the FTP client was unable to build a list of filenames 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.

User or Operator Response: None.

System Programmer Response: Verify adequate storage available for memory files.

Source Data Set: 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 aborted and the file transfer is completed using base processing. Any MVS load modules transferred will not be executable on the target system.

User or Operator Response: Contact your system programmer.

System Programmer Response: Ensure that the server and client are both Communications Server for OS/390 at the V2R10 level or higher. If not, contact the vendor of the non-IBM FTP program for service.

Source Data Set: 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 aborted, and no files will be transferred. FTP waits for the next subcommand.

User or Operator Response: Contact your 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.

Source Data Set: 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 aborted, and no files will be transferred. FTP waits for the next subcommand.

User or Operator Response: Contact your 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Contact your system programmer

System Programmer Response: Use the *z/OS DFSMSdfp Utilities* manual to determine the corrective action for the IEBCOPY return code.

Source Data Set: 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.

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

System Programmer Response: Correct the data set name in the FTP.DATA file. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Send the Lrecl X data set using the correct FILETYPE, MODE, STRUCT, TYPE, or ASATRANS setting.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect an FTP Client trace that includes the failure and call the IBM Support Center.

Source Data Set: 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.

User or 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: None.

Source Data Set: EZAFTPCL

Procedure Name: debug

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 HFS 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.

User or 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, refer to the *z/OS Communications Server: IP Configuration Reference* for information about configuring the FTP client for SOCKS.

Source Data Set: 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 HFS 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

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.

User or Operator Response: None.

System Programmer Response: None.

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: None.

User or Operator Response: Activate the FTP client and server traces as follows:

```
SITE DEBUG=(NONE,ACC,SOC(2))
DEBUG NONE ACC SOC(2)
```

Retry the subcommand 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPCL

Procedure Name: locstat

EZA2892I Secure port 990 does not support the -a or -r start parameter

Explanation: You specified the secure port 990. The **-a** or **-r** start parameters are not allowed when you specify port 990.

System Action: FTP ends.

User or Operator Response: Start FTP without the **-a** or **-r** start parameter if the secure port is used, or use the **-a** or **-r** start parameter with another port number.

System Programmer Response: None.

Source Data Set: EZAFTPCY

Procedure Name: processStartParms

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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Correct the FTP.DATA file to 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.

Source Data Set: EZAFTPCZ

Procedure Name: ftpOpen

EZA2897I Authentication negotiation failed

Explanation: The client was unable to negotiate authentication with the server.

System Action: FTP continues.

User or Operator Response: Activate the FTP client and server traces as follows:

- SITE DEBUG=(NONE,ACC,SOC(2))
- DEBUG NONE ACC SOC(2)

Retry the connection 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.

Source Data Set: 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.

User or Operator Response: Activate the FTP server and client traces as follows:

- SITE DEBUG=(NONE,ACC,SOC(2))
- DEBUG NONE ACC SOC(2)

Retry the connection 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Security Server Network Authentication Service Programming* for an explanation of the gss_import_name() function.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Security Server Network Authentication Service Programming* for an explanation of the gss_init_sec_context() function.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Security Server Network Authentication Service Programming* for an explanation of the gss_wrap() function.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Security Server Network Authentication Service Programming* for an explanation of the gss_unwrap() function.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Security Server Network Authentication Service Programming* for an explanation of the major status code.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Security Server Network Authentication Service Programming* for an explanation of the minor status code.

Source Data Set: 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.

User or Operator Response: Increase the SECURE_PBSZ value in the FTP.DATA file.

System Programmer Response: None.

Source Data Set: EZAFTPFC

Procedure Name: gss_receive_data

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.

User or 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: None.

Source Data Set: EZAFTPCZ

Procedure Name: ftpOpen

EZA2918I Filetype SQL requires type A (ASCII) or E (EBCDIC)

Explanation: The user issued a PUT or APPEND subcommand. The filetype 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.

User or Operator Response: Change the data type by issuing the ASCII or EBCDIC subcommand and reissue the put or append subcommand.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: EZAFTPCL

Procedure Name: debug

EZA2927I SJISKANJI (NOTYPE requires type A (ASCII)

Explanation: A 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.

User or Operator Response: Issue the Type A command prior to entering the SJISKANJI (NOTYPE command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the binary or TYPE I subcommand and reenter the transfer request.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the ASCII or TYPE A subcommand and reenter the transfer request.

System Programmer Response: None.

Source Data Set: 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 codesets 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.

User or Operator Response: Use LOCSITE SBDATACONN to build or select a translation table that can translate the codepoints in the data.

System Programmer Response: None.

Source Data Set: 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 codesets 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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). 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the format and syntax as indicated and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or Operator Response: Resubmit the request when a socket is available.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Resubmit the command when storage is available to create the socket.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

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

System Programmer Response: Check the host to determine why no port can be allocated.

Source Data Set: 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.

User or 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 Reference*.

Source Data Set: 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.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or 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 Application Programming Interface Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or 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 Application Programming Interface Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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 Reference* .

System Action: TCPIP continues.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: Correct the specification and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: Correct the syntax and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MAIN

Procedure Name: main

Chapter 4. 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.

User or 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 Reference*.

Source Data Set: 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.

User or 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 Reference*.

Source Data Set: 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.

User or 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 Reference*.

Source Data Set: 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.

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

System Programmer Response: Correct the options as necessary and recompile.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or Operator Response: Correct the NIDL command and resubmit it.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or Operator Response: Resubmit the command without specifying the ansi option.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 Reference*.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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 Reference*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Respond as indicated by the *status* displayed in this message.

Source Data Set: 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.

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

System Programmer Response: Respond as indicated by the *status* displayed in this message.

Source Data Set: 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.

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

System Programmer Response: Respond as indicated by the *status* displayed in this message.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 Application Programming Interface Guide*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

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

System Programmer Response: Respond as indicated by the *status* portion of this message.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Correct the data set as indicated and resubmit it for processing.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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*.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RPC@UTIL

Procedure Name: rpc__dump_pkt_hdr

EZA3950I **NDB Port Manager for MVS Version 3 started**

Explanation: This message is printed to job log output when the NDB port manager is started. It displays the current level, version, and release of code being used.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDBPS

Procedure Name: NDBPS

EZA3951E **Cannot create** *type* **service.**

Explanation: NDB uses remote procedure call (RPC) to handle requests between the NDB port manager, NDB servers and NDB clients. NDBPS was trying to establish the indicated type of service via RPC, but it failed.

System Action: The NDB port manager ends.

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

System Programmer Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure that the portmapper is started and restart the procedure PORTS.

Source Data Set: NDBPS

Procedure Name: NDBPS

EZA3952E Unable to register (PORTS_PROG, PORTS_VERS, *string*).

Explanation: NDB uses remote procedure call (RPC) to handle requests between the NDB port manager, NDB servers and NDB clients. NDBPS was trying to register either a TCP or UDP service transport handle, as indicated by *string* with the local Portmapper via the RPC call svc_register, but it failed.

System Action: The NDB port manager ends.

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

System Programmer Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure the portmapper is started and restart the procedure PORTS.

Source Data Set: NDBPS

Procedure Name: NDBPS

EZA3953E Unable to free arguments

Explanation: NDB uses remote procedure call (RPC) to handle requests between the NDB port manager, NDB servers and NDB clients. NDBPS was trying to free storage previously allocated for argument decoding via the RPC call svc_freeargs, but it failed.

System Action: The NDB port manager ends.

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

System Programmer Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure the Portmapper is started and restart the procedure PORTS.

Source Data Set: NDBPS

Procedure Name: NDBPS

Chapter 5. EZA4*xxxx* messages

EZA4000I PORTCLNT for MVS Version 3 started

Explanation: This message is printed to job log output when the NDB port client first starts to indicate that the NDB port client is started and to display the current host, level, and version of code being used.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4001E Parameters to be specified are: HOMEID USERID DB2SSID NUMSRV TRACE

Explanation: This message is printed to job log output when the NDB port client finds it has an incorrect number of startup parameters. It lists the possible parameters that PORCLNT will accept. The first 3 are required. If NUMSRV is not specified, it is set to 1, and PORTCLNT continues. TRACE is an optional parameter and is usually turned off.

System Action: NDB port client startup terminates.

User or Operator Response: Correct the parameters and restart the procedure PORTC.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4002E MALLOC failed for storage for SSCB (PCb) fields

Explanation: PORTCLNT tried to obtain storage for an SSCB type of internal control block but was unsuccessful.

System Action: NDB port client startup terminates.

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

System Programmer Response: Increase the region size and restart the application.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4003E Caller is neither the NDB Server nor NDB Client. Exit(4).

Explanation: PORTCLNT acknowledges communications from two places, one of the NDB servers or from an NDB client. A communication has been received that did not come from one of these sources.

System Action: NDB port client startup terminates.

User or Operator Response: Make sure that NDB is properly installed. If the error persists, notify the system programmer.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4004E DB2 subsystem name, *name*, is longer than is allowed by DB2.

Explanation: The parameter DB2SSID contained a DB2 subsystem name that was longer than 4 bytes.

System Action: NDB port client startup terminates.

User or Operator Response: Verify the DB2 subsystem name, correct it on the JCL, and restart the procedure PORTC.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4005E NUMSRV contains an incorrect value. value is: *value*

Explanation: The parameter NUMSRV contained a value other than a number 1 through 20. This message displays the value PORTCLNT received for the parameter NUMSRV. The parameter is set to 1 and processing continues.

System Action: The NDB port client continues.

User or Operator Response: Correct the parameter and restart the procedure PORTC.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4006E NUMSRV is not between 1-20. defaulting to 1.

Explanation: The parameter NUMSRV contained a value other than a number 1 through 20; the value in NUMSRV is reset to 1 by PORTCLNT.

System Action: The NDB port client continues.

User or Operator Response: Correct the value of NUMSRV and restart the procedure PORTC.

System Programmer Response: Assist the user as necessary.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4007I Number of NDB Servers being started is *number*

Explanation: This message is printed to job log output during NDB port client startup. It indicates the number of NDB servers being started by PORCLNT.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4008E MTF function '*function***' failed. RC is** *rc*

Explanation: PORTCLNT uses the C/370™ Multitasking Facility to startup multiple NDB servers. While trying to execute the indicated MTF function an error was encountered, as specified by *rc* portion of this message.

System Action: The NDB port client ends.

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

System Programmer Response: Use the return code displayed in this message and the list of return codes in the *C/370 Programming Guide* or the header file MTF.H to determine the cause of the error, and respond as indicated.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4009E RPC error occurred during ports_msg_1 call. Exit(7).

Explanation: NDB uses remote procedure call (RPC) to handle requests between NDB clients and NDB servers. It is also used by the NDB port client to register each NDB server started with the NDB port manager (the NDB component that assigns NDB servers to clients). During the RPC call from the NDB port client to the NDB port manager to register this NDB server, an RPC failure occurred.

System Action: The NDB port client ends.

User or Operator Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure the Portmapper and NDB port manager are started and restart PORTC.

System Programmer Response: Assist the user as necessary.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4010E All the ports are busy. Please try again later.

Explanation: The NDB port manager can handle up to 20 NDB servers. This is because it has 20 program numbers reserved in the ETCRPC member of the SEZAINST data set. These program numbers become associated with a port when an NDB client requests an NDB server and one is assigned. Currently, all ports are in use.

System Action: The NDB port client ends.

User or Operator Response: Wait for a free port before trying to establish a connection.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4011I Server *number* **started. Prognum is** *program number***(Hex),** *program number***(Dec).**

Explanation: This message is printed to job log output when the NDB port client has started up an NDB server. The program number from the ETCRPC member of the SEZAINST data set assigned to this NDB server is reported in hexadecimal and decimal.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4012I PORTCLNT exit. Normal shutdown completed.

Explanation: This message is printed to job log output when the NDB port client is stopped.

System Action: PORTCLNT ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PORTCLNT

Procedure Name: PORTCLNT

EZA4150I NDB Server *program number* **(HEX) started, accessing** *name*

Explanation: This message is printed to job log output when an NDB server is started by the NDB port client. The program number from the ETCRPC member of the SEZAINST data set assigned to this NDB Server is reported in hexadecimal. The DB2 subsystem that can be accessed by this NDB Server is specified in *name*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDBSS

Procedure Name: NDBSS

EZA4151I *host* **NDB Server received a call from host userid** *userid*

Explanation: This message is printed to job log output when an NDB server is called by an NDB client. *host* is MVS or VM, indicating the version of NDB server code being used and the host user id provided by the user of the NDB client is also displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4152E MALLOC of *string* **failed in function** *string*

Explanation: NDBSS tried to obtain storage for whatever is specified in the first string but was unsuccessful. The function trying to do the MALLOC is named in the second string.

System Action: The NDB server terminates the current request.

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

System Programmer Response: Increase the region size for the indicated function.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4153E Invalid NDBC control block. Reason: *string*.

Explanation: The NDB server and the NDB client use a control block, NDBC, to transfer information to each other. Several fields of NDBC are used as validity checks also. This message indicates that one of these fields contains an incorrect value.

If string Is:	Bad field Is:	Should be:
NDB Release numbers don't match	NDBREL	1
NDB Version numbers don't match	NDBVER	2
Eyecatcher not for NDBC control block	NDBCB	NDBC
NDB Server names don't match	NDBSNAME	netdbsrv

System Action: The NDB server terminates the current request.

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

System Programmer Response: Get the member NDBCLTC from the SEZAINST data set again and rename it according to the client platform. For more information, see the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4154E Cannot create *type* service.

Explanation: NDB uses remote procedure call (RPC) to handle requests between NDB clients and NDB servers. NDBSS was trying to establish either a TCP or UDP (as specified by *type*) service transport handle via RPC, but it was unsuccessful.

System Action: The NDB server terminates.

User or Operator Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure the Portmapper and NDB port manager are started and restart PORTC.

System Programmer Response: Assist the user as necessary.

Source Data Set: NDBSS

Procedure Name: NDBSS

EZA4155E Unable to register (NETDB_PROG, NETDB_VERS, *type*).

Explanation: NDB uses remote procedure call (RPC) to handle requests between NDB clients and NDB servers. NDBSS was trying to register either a TCP or UDP service transport handle, as indicated by *type*, with the local portmapper via the RPC call svc_register, but it was unsuccessful.

System Action: The NDB server terminates.

User or Operator Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure the portmapper and NDB port manager are started and restart PORTC.

System Programmer Response: Assist the user as necessary.

Source Data Set: NDBSS

Procedure Name: NDBSS

EZA4156E Error occurred for SQL CLOSE CURSOR *string*. SQLCODE is *number*.

Explanation: The indicated function encountered an error while doing an SQL CLOSE CURSOR statement. The type of error is indicated by the SQLCODE displayed in the message.

System Action: The NDB Server continues processing

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

System Programmer Response: Use the SQLCODE displayed in the message and the list of codes in *DB2 Messages and Codes* to determine the cause of the error, and respond as indicated.

Source Data Set: NDBSS

Procedure Name: DBUTIL2

EZA4157E Error occurred for SQL *string*. SQLCODE is *number*.

Explanation: A DB2 error condition was encountered while processing an SQL statement. The SQL statement and the function reporting the error are indicated by *string*, and the DB2 error code reported in the SQLCA field SQLCODE is indicated by *number*.

System Action: The NDB Server terminates the current request.

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

System Programmer Response: Use the SQLCODE displayed in the message and the list of codes in *DB2 Messages and Codes* to determine the cause of the error and respond as indicated.

Source Data Set: NDBSS

Procedure Name: DBUTIL2

EZA4158E Single row of query data is larger than reply buffer. Query ended.

Explanation: The length of one row of data to be returned from the current query is greater than the length of the NDB reply buffer. The size of the NDB reply buffer is specified in the NDBC field NDBRPDLN.

System Action: The NDB Server terminates the current query.

User or Operator Response: Retrieve fewer columns per query either by issuing multiple queries or by making the query more restrictive in the number of columns requested, or increase the size of the NDB reply buffer. For more information on increasing the size of the reply buffer, see the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NDBSS

Procedure Name: DBUTIL2

EZA4159E Unable to free arguments

Explanation: NDB uses remote procedure call (RPC) to handle requests between NDB clients and NDB servers. NDBSS was trying to free storage previously allocated for argument decoding via the RPC call svc_freeargs, but it failed.

System Action: The NDB Server terminates.

User or Operator Response: Check job log output for related RPC error messages. Correct any errors reported. Make sure the portmapper and NDB port manager are started and restart PORTC.

System Programmer Response: Assist the user as necessary.

Source Data Set: NDBSS

Procedure Name: NDBSS

EZA4160E Host userid or password is not valid

Explanation: The user id or password supplied to the NDB server from the NDB client on behalf of the user was not accepted by the security facility (such as RACF).

System Action: The NDB Server terminates the current request.

User or Operator Response: Make sure that you are specifying a valid host user id and password when you invoke the NDB client.

System Programmer Response: Assist the user as necessary.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4161E CAF error: Wrong number of parms or end-of-list bit off

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making one of these calls, CAF reported to NDB that an error has occurred. The error was that an incorrect number of parameters was specified on the call to CAF by NDB or that the end-of-list bit, which should have been on, was set to off.

System Action: The NDB server terminates the current request.

User or Operator Response: Make sure that the Call Attachment Facility is properly installed and DB2 is started, and restart your NDB Client session. If the error persists, notify the system programmer of the error.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4162E CAF error: Unrecognized function parameter

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making one of these calls, CAF reported to NDB that an error has occurred. The error was that the service NDB requested of CAF was not a supported function, such as OPEN or CLOSE.

System Action: The NDB Server terminates the current request.

User or Operator Response: Make sure that the Call Attachment Facility is properly installed and that DB2 is started. If the error persists, notify the system programmer.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4163E CAF error: Access two different DB2 subsystems, one TCB

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making one of these calls, CAF reported to NDB that an error has occurred. The error was that CAF received two calls from one NDB server, each call requesting to access a different DB2 subsystem.

System Action: The NDB Server terminates the current request.

User or Operator Response: Make sure that the Call Attachment Facility is properly installed and that DB2 is started. If the error persists, notify the system programmer.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4164E CAF error: Release level mismatch between DB2 and CAF

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, thereby also establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making one of these calls, CAF reported to NDB that an error has occurred. The error was that the level of DB2 currently executing and the level of CAF currently executing are not the same.

System Action: The NDB server terminates the current request.

User or Operator Response: Make sure the Call Attachment Facility is properly installed and that DB2 is started and restart the NDB client session. If the error persists, notify the system programmer.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4165E CAF system error: Probable error in the attach or DB2

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, thereby also establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making one of these calls, CAF reported to NDB that an error has occurred. The error was that CAF detected an error it could not identify.

System Action: The NDB server terminates the current request.

User or Operator Response: Make sure the Call Attachment Facility is properly installed and that DB2 is started and restart the NDB client session. If the error persists, notify the system programmer.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4166E CAF *string* DB failed. Return code = *number*. Reason code = *hex number*(Hex)

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, thereby also establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making one of these calls, CAF reported to NDB that an error has occurred.

System Action: The NDB server terminates the current request.

User or Operator Response: Make sure the Call Attachment Facility is properly installed and that DB2 is started and restart your NDB client session. If the error persists, notify the system programmer.

System Programmer Response: Use the codes displayed in this message and *DB2 Messages and Codes* to determine the cause of the error and respond as indicated.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4167E NDB server could not do CAF OPEN DB; DB2

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. The NDB server was unable to make a CAF OPEN call, because the name of the DB2 subsystem to be used was missing.

System Action: The NDB server terminates the current request.

User or Operator Response: Make sure that the DB2 subsystem name was provided when PORTC JCL was started. Check the PORTC job log console for previous error messages. If the error persists, notify the system programmer.

System Programmer Response: Obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4168E Load of CAF modules failed. Return code is *rc*.

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While trying to load the CAF modules needed to make these calls, an error occurred because DB2 was unable to load the modules.

System Action: The NDB server terminates the current request.

User or Operator Response: Make sure the Call Attachment Facility is properly installed and that DB2 is started, and restart your NDB client session. If the error persists, notify the system programmer.

System Programmer Response: Use the return code displayed in this message and *DB2 Messages and Codes* to determine the cause of the error and respond as indicated.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4169E Privilege to execute plan EZAND320 not granted to this user or to public

Explanation: During NDB installation, a job, NDBSETUP, was to have been run that would bind a DBRM to DB2 that is used by NDB and is called EZAND320. NDBSETUP also granted the DB2 execute, or run, privilege to all users (PUBLIC) on EZAND320. If a customer chose to restrict use of NDB, they could modify NDBSETUP to grant to only a

select set of user IDs or could have done the bind of EZAND320 and grant to select users through DB2 facilities. The current user is not authorized to use EZAND320, and therefore cannot use NDB.

System Action: The NDB server terminates the current request.

User or Operator Response: Contact the system administrator for authorization to use DBUTIL2.

System Programmer Response: Grant access to DBUTIL2 in accordance with current internal guidelines.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4170E DB2 Subsystem is unavailable. DB2 not active at OPEN DB.

Explanation: The NDB servers use the DB2 Call Attachment Facility (CAF) to establish a connection with DB2. NDB uses two services of CAF, one to open the plan in DB2 used by NDB and called EZAND320, establishing a connection to DB2; and another to close the plan EZAND320 and end the connection. While making an OPEN call, CAF reported to NDB that the DB2 subsystem specified in the call was not active.

System Action: The NDB server terminates the current request.

User or Operator Response: Start the DB2 subsystem that was specified in the DB2SSID parameter at PORTC startup.

System Programmer Response: Assist the user as necessary.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4171E 'Should never occur' error number *number* **has occurred.**

Explanation: This message indicates a possible programming error.

System Action: The NDB server terminates the current request.

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

System Programmer Response: Make sure that DB2, TCP/IP, and NDB are installed correctly. If so, obtain trace information and contact the IBM Software Support Center.

Source Data Set: NDBSS

Procedure Name: NDBSRV

EZA4172E SQLCA error tokens are *string*

Explanation: This message follows message EZA4156E or EZA4157E when the SQLCA contains error tokens. The SQLCA error tokens are displayed.

System Action: The NDB server terminates the current request.

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

System Programmer Response: Use the SQLCA tokens, the SQLCODE from the previous message, and the list of codes in the *DB2 Messages and Codes* to determine the cause of the error. Respond as indicated.

Source Data Set: NDBSS

Procedure Name: DBUTIL2

EZA4173E Error during host userid/password validation. SAF RC is *rc***, security RC is** *rc***, security reason code is** *code***.**

Explanation: The installed host security system detected an error while attempting to validate the specified userid and password. The return code from the MVS Security Access Facility (SAF), as well as the host security system return code and reason code, are displayed.

System Action: The NDB server terminates the current NDB session.

User or Operator Response: Make sure that you are specifying a valid host user ID and password when you invoke the NDB client.

System Programmer Response: Locate the specified return codes and reason code in your installed host security system's *Messages and Codes* manual to determine the cause of the error. Respond as indicated. For RACF, refer to the *z/OS Security Server RACF Messages and Codes*.

Source Data Set: NDBSS

Procedure Name: NDBSRV

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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: portmap

Procedure Name: main(), svc_run()

EZA4257I PORTMAPPER STOPPED

Explanation: The Unix System Services Portmapper issues this message just before it exits in response to a stop command from the operator.

System Action: The portmapper stops.

User or Operator Response: None.

System Programmer Response: Restart the portmapper as needed.

Source Data Set: portmap

Procedure Name: main()

EZA4275E auth_none.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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: AUTH@UNI

Procedure Name: marshal_new_auth()

EZA4276E get_myaddress&colon 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check for the correct entries declared in the parameters of the svctcp_create(), getsockname(), and listen() calls, and restart the program. Refer to *z/OS Communications Server: IP Programmer's Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or 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, refer to *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.

User or Operator Response: Correct the syntax and restart.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Change the output file name and restart.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: None. Correct the program.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: None. Correct the program.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: None. Examine the program data for an unterminated string or an incorrect pointer. Correct the program.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: If the compiler does not provide a reset to *nfiles*, add a software reset to *nfiles*.

Source Data Set: 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.

User or Operator Response: Correct the source file and restart.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the source file and restart.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the source file and restart.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: Correct the program.

Source Data Set: 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.

User or Operator Response: Correct the input program and restart; or tell the system programmer about the error.

System Programmer Response: Correct the input program.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: Correct the program.

Source Data Set: 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.

User or Operator Response: Correct the program and restart; or tell the system programmer about the error.

System Programmer Response: Correct the program.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: RPCINFO

Procedure Name: udpping()

EZA4327I Program *number* version *version* is not available

Explanation: The indicated program is not ready.

System Action: The system continues.

User or Operator Response: Start the server that uses the required program number.

System Programmer Response: None.

Source Data Set: RCPINFO

Procedure Name: Pmapdump()

EZA4328I Program *number* version *version* ready and waiting

Explanation: The indicated program is ready.

System Action: The system continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Start a server that registers a procedure on the Portmapper.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: PING the host. Determine whether the host name is correct and whether the name server is running.

System Programmer Response: None. Make any necessary changes to the *hlq*.HOSTS.LOCAL data set and run MAKESITE.

Source Data Set: 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.

User or 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 Reference*.

System Programmer Response: None.

Source Data Set: 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.

User or 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).

Source Data Set: 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.

User or Operator Response: Inform the system programmer of this message.

System Programmer Response: Following are the possible responses for this message:

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.

EZA4340E

Message	Response
• 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 Application Programming Interface Guide* 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. TCP and UDP are the supported transports.
• 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 B, "System Return Codes," 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*.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 Reference* for more information.

Source Data Set: 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.

User or 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 Reference* for more information.

Source Data Set: 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.

User or 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 Reference* for more information.

Source Data Set: 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.

User or 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 Reference* for more information.

Source Data Set: 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.

User or Operator Response: None.

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 Reference* for more information.

Source Data Set: 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.

User or 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 Reference* for more information.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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. gethostbyaddr() failed.

System Action: Processing continues.

User or Operator Response: None.

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.

Source Data Set: MVRSHD

Procedure Name: doit

EZA4381I Accept *socket number* **from** *address* **on** *port*

Explanation: The server has accepted a connection from the indicated socket, address, and port number.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVRSHD

Procedure Name: main

EZA4382I Connecting on *socket number* **to** *address port*

Explanation: The server is opening a connection on the socket number to the specified IP address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVRSHD

Procedure Name: main.

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 userid of the client system.

System Programmer Response: None.

Source Data Set: MVRSHD

Procedure Name: doit

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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or 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: None.

Source Data Set: MVRSHD

Procedure Name: Process_options

EZA4399E Invalid trace option *value*

Explanation: An incorrect option has been specified for the TRACE= parameter. The following is a list of the valid options for this parameter:

LOG/NOLOG
SEND/NOSEND
CLIENT=*client*
ALLCLIENTS
RESET

System Action: Processing continues.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVRSHD

Procedure Name: process_options

EZA4402E Invalid option *option*

Explanation: An incorrect option has been specified for the TRACE= parameter. The following is a list of 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Decrease the size of the command to 246 characters or less.

Source Data Set: 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.

User or Operator Response: Determine which jobnames are duplicated and purge the duplicate job output.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Determine which jobnames are duplicated and purge the duplicate job output.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Wait and then reissue the REXEC request.

System Programmer Response: None.

Source Data Set: 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 userid of the submitter of the request.

System Action: Processing continues.

User or Operator Response: Wait and then reissue the REXEC request.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Wait and then reissue the REXEC request.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 userid 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.

User or 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*.

Source Data Set: MVRSHD

Procedure Name: main

EZA4422E rexecd: Permission Denied.

Explanation: Either the userid or the password specified for the userid is not valid on the remote system.

System Action: Processing continues.

User or Operator Response: Check the -l and -p parameters on REXEC and make sure that they are correct. If the userid or the password, or both are being retrieved from a NETRC file, check the userid and password in the NETRC file.

System Programmer Response: None.

Source Data Set: MVRSHD

Procedure Name: main.

EZA4423I Parameters: PURGE=*purge***,PREFIX=***prefix*

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 will be a sequential number between 1 and MAXCONN.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the option with a correct value.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If new PREFIX value is required then stop the server and start it again with the new PREFIX value.

System Programmer Response: None.

Source Data Set: 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.

User or 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 C/C++ Run-Time Library Reference* for more information about the fetch() command.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVRSHD

Procedure Name: process_options

EZA4726I Usage: rexec -? -d -l *usr* -p *pwd* -n -s *port* -t *fn* fhost cmd

EZA4727I options: -

EZA4728I -? display this message.

EZA4729I -d turn on debug tracing.

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

Explanation: This group of messages is posted when REXEC is entered without any parameters, or with the -? parameter, or under certain conditions that are not valid.

System Action: REXEC ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: usage

EZA4736I close error con # *errconn*

Explanation: An error has occurred on the port listed and the port is being closed.

System Action: REXEC ends.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Message EZA4779I precedes this message and explains what needs to be corrected prior to rerunning the job.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: close

EZA4738I getnextnote until DD

Explanation: The program has entered the procedure recv_notices.

System Action: REXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Varies according to the message displayed.

System Programmer Response: Varies according to the message displayed.

Source Data Set: 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.

User or Operator Response: Varies according to the message displayed.

System Programmer Response: Varies according to the message displayed.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Varies depending on the message EZA4744I.

System Programmer Response: Varies depending on the message EZA4744I.

Source Data Set: 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.

User or Operator Response: Varies depending on the message EZA4744I.

System Programmer Response: Varies depending on the message EZA4744I.

Source Data Set: 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 shut down.

System Action: REXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter a command.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: prompt_for_command

EZA4753I PASSWD (*fhost: userid*)

Explanation: A password needs to be entered for the userid on the remote host. If quit is entered, REXEC will end.

System Action: REXEC awaits input.

User or Operator Response: Enter the password.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the user ID.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the remote host name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the user_id.NETREC.DATA data set.

System Programmer Response: None.

Source Data Set: 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

User or 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: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a correct logon ID.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid password.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid port number.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with valid parameters.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid host name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the complete REXEC command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: REXEC_util

EZA4775I sending *string*

Explanation: The port number and string that has been sent to the remote host is displayed.

System Action: REXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reenter the command using a valid remote host name.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Varies depending on the message displayed.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Varies depending on the message displayed.

Source Data Set: REXEC

Procedure Name: REXEC_Client

EZA4788I active open complete on port *connection*

Explanation: An active open is complete.

System Action: REXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: REXEC_Client

EZA4789I REXEC complete

Explanation: The REXEC program has ended.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the quoted string and reenter the command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: REXEC_Client

EZA4801I MVS TCP/IP REXEC *version*

Explanation: The version of REXEC is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: GETPARM

EZA4802I -t 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.

System Action: REXEC ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Correct or enter the required parameter.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Verify that the data set that has been specified with the -t option is correctly named and available to the system.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Correct the port parameter.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Correct the port number.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reissue the REXEC command. If the problem persists, seek assistance from appropriate system support personnel.

System Programmer Response: None.

Source Data Set: 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 diplayed.

System Action: REXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: REXEC

Procedure Name: from_netrc

EZA4986I close error con # *connection*

Explanation: An error has occurred on the port listed and the port is being closed.

System Action: RSH ends.

User or Operator Response: Message EZA5029I precedes this message and explains the correction that needs to be made prior to rerunning the job.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: RSH

Procedure Name: close

EZA4988I getnextnote until DD

Explanation: The program has entered the procedure recv_notices.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

EZA4990I

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.

User or Operator Response: Varies according to the message displayed.

System Programmer Response: Varies according to the message displayed.

Source Data Set: 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.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Varies depending on the message displayed.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Varies depending on message EZA4994I.

System Programmer Response: Varies depending on message EZA4994I.

Source Data Set: 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.

User or Operator Response: Varies depending on information displayed in message EZA4994I.

System Programmer Response: Varies depending on information displayed in message EZA4994I.

Source Data Set: 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 shut down.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: recv_notices

Chapter 6. EZA5*xxxx* messages

EZA5000I unknown notification

Explanation: The connection state message is not one that has been previously defined.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: show_variables

EZA5008I USERID: *userid*

Explanation: The DEBUG option has been specified and the logonid is displayed.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid logon ID.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid port number.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with valid parameters.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: GETPARM

EZA5025I sending *string*

Explanation: This is the port number and the string that has been sent to the foreign host.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP*
| *Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

| **System Programmer Response:** Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reenter the command using a valid foreign host name.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Varies depending on the message displayed.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP*
| *Messages Volume 1 (EZA)*.

System Action: Varies depending on the message displayed.

User or Operator Response: Varies depending on the message displayed.

| **System Programmer Response:** Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Varies depending on the message displayed.

System Programmer Response: Varies depending on the message displayed.

Source Data Set: RSH

Procedure Name: Rsh_client

EZA5038I active open complete on port *connection*

Explanation: An active open is complete.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: Rsh_client

EZA5042I Usage: rsh -? -d -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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid host name.

System Programmer Response: None.

Source Data Set: 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

User or Operator Response: Reenter the command with a valid host name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: rsh_util

EZA5048I RSH complete

Explanation: The program has ended.

System Action: RSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: Rsh_Client

EZA5049I locuser *locuser*

Explanation: This message displays the local user id.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSH

Procedure Name: show_variables

EZA5125I IBM MVS SMTP *version* **on** *date*

Explanation: Introduces SMTP and displays the version level and the date.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5127I Jobname of SMTP Server: *name*

Explanation: Displays the name of the Simple Mail Transfer Protocol job. It is usually SMTP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5128I TCP Network Domain name: *name*

Explanation: Displays the name of the local host.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5129I Gateway TCP Network to NJE: *path*

Explanation: Displays whether TCPIP will gateway mail to and from remote NJE systems.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5130I NJE Network Node Name: *name*

Explanation: Displays the name of the NJE node on which the SMTP is running. To change this name, change the SMTPNODE variable in the *hlq*.SEZAINST(SMTPNOTE). For information on customizing this and other members, and for information on the SMTPNODE variables, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStream

EZA5131I NJE Domain Name: *domain name*

Explanation: Displays the name of the NJE domain if SMTP is running as a mail gateway. To change this name, edit the NJEDOMAIN statement in *hlq*.SMTP.CONFIG. The default name is the null string. For more information on the NJEDOMAIN statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStream

EZA5133I Local Delivery Format: *punch / netdata*

Explanation: Displays the spool data set format for mail delivery to recipients on the local host. To change the format, change the LOCALFORMAT statement in *hlq*.SMTP.CONFIG. There are 2 choices for the format: NETDATA and PUNCH. The default format is NETDATA. For more information on the LOCALFORMAT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStream

EZA5134I NJE Delivery Format: *punch / netdata*

Explanation: Displays PUNCH or NETDATA as options. Specifies how long the records may be. The PUNCH parameter displayed means that records can be at most 80 characters while the default format, NETDATA, allows records longer than 80 characters and they arrive as MESSAGE-type records. To change the setting, change the NJEFORMAT statement in *hlq*.SMTP.CONFIG. For more information on the NJEFORMAT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStream

EZA5136I Postmaster Address: *address*

Explanation: Displays the current address where SMTP delivers all mail addressed to POSTMASTER. To change the address, change the POSTMASTER statement in *hlq*.SMTP.CONFIG. The default postmaster is TCPMAINT. For more information on the POSTMASTER statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5137I Userid for Bad Spool Files: *user id*

Explanation: Displays the user id that receives all the unreadable spool files and looping mail. To change the user id, change the BADSPOOLFILEID statement in *hlq*.SMTP.CONFIG. The default user id is TCPMAINT. For more information on the BADSPOOLFILEID statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5139I Port for Server SMTP: *server port*

Explanation: Displays the port the SMTP utilizes. The port number 25 is normally reserved in *hlq*.PROFILE.TCPIP. To change the port that SMTP uses, you must change the PORT statement in *hlq*.SMTP.CONFIG and, unless testing, change the data set *hlq*.PROFILE.TCPIP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5140I Inactivity Timeout: *seconds* **seconds**

Explanation: Displays the number of seconds until SMTP considers the connection dead. At that time, SMTP closes the connection. To change the amount of time, change the INACTIVE statement in *hlq*.SMTP.CONFIG. The default amount of time for the INACTIVE statement is 180 seconds. For more information on the INACTIVE statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5141I Finish Open Timeout: *seconds* **seconds**

Explanation: Displays the number of seconds SMTP waits while trying to establish a connection to a remote site. After the given amount of time, SMTP ends the connection. To change this setting, change the FINISHOPEN statement in *hlq*.SMTP.CONFIG. The default amount of time for the FINISHOPEN statement is 120 seconds. For more information on the FINISHOPEN statement, see the *z/OS Communications Server: IP Configuration Reference* .

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStream

EZA5142I Retry down sites every: *minutes* **minutes**

Explanation: Displays the number of minutes SMTP waits between attempts to deliver mail to a host that is down. It is adjustable by using the RETRYINT statement in *hlq*.SMTP.CONFIG. The default time for the RETRYINT statement is 20 minutes. For more information on the RETRYINT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5143I Return mail older than: *days* **days**

Explanation: Displays the number of days after which SMTP returns mail as undeliverable. SMTP returns the mail to the sender with a note listing recipients that did not receive the mail. To change the number of days, change the RETRYAGE statement in *hlq*.SMTP.CONFIG. The default number of days for the RETRYAGE statement is 3 days. For more information on the RETRYAGE statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5144I Max Length of Accepted Mail: *number* **bytes**

Explanation: Displays the maximum size, in bytes, of mail that is accepted over a TCP connection. To change the maximum size, change the MAXMAILBYTES statement in *hlq*.SMTP.CONFIG. The default for MAXMAILBYTES is 524 288 (512K) bytes. For more information on the MAXMAILBYTES statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5145I Translate Table : *data set name* **/ Compiled-in. Unable to load:** *data set name*

Explanation: If SMTP is successful in loading the translation table data set, then the data set name is loaded and displayed. Otherwise, the other error message is displayed. See the *z/OS Communications Server: IP Configuration Reference* for information about managing SMTP translation tables.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5156E Unrecognized option in configuration file: *option*

Explanation: The indicated option is an incorrect configuration data set option. See the *z/OS Communications Server: IP Configuration Reference* for information about the the valid options.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration data set, and try again.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5157E Invalid LOG file destination: *destination*

Explanation: The LOGFILE DD statement is incorrect or absent. SMTP does not log the session.

System Action: SMTP continues.

User or Operator Response: Verify the presence of the LOGFILE DD statement in *hlq*.SEZAINST(SMTPPROC) and make sure the data set destination is correct. For more information on the LOGFILE DD statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the user as necessary. None.

Source Data Set: SMTP

Procedure Name: Error

EZA5158E Invalid Address in Restricted List: *address*

Explanation: SMTP found an incorrect address in the RESTRICT list. Addresses should be in the form *user_id@node_id*, with the asterisk character (*) representing wild cards.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration data set, and try again.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5159E Invalid Local Delivery Format: *format*

Explanation: An incorrect delivery format for the LOCALFORMAT option in the configuration data set was specified. The only valid delivery formats are NETDATA and PUNCH.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5160E Invalid NJE Delivery Format: *format*

Explanation: An incorrect option for the NJEFORMAT option in the configuration data set was specified. The only valid choices for this option are NETDATA or PUNCH.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5161E Invalid Disposition in Restrict command: *command*

Explanation: The disposition specified with the RESTRICT option is incorrect. Use only the valid options of PURGE, RETURN, and TRANSFERTO with the RESTRICT option.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5162E Invalid Address for Mailer: *address*

Explanation: The length of the user ID for the receiving address exceeds 8 characters, which is the maximum allowable MVS user ID length for the MAILER option.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5163E Invalid Max Mail Bytes: *parameter*

Explanation: The indicated parameter for the MAXMAILBYTES option was not a positive integer between 1 and 2 147 483 647.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5164E Invalid Post Master Userid: *userid*

Explanation: The indicated user ID does not exist for the POSTMASTER option in the configuration data set.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Check that this user ID exists. Otherwise, correct the SMTP configuration data set, specifying a valid user ID, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5165E Invalid Badspoolfileid Userid: *userid*

Explanation: The reason for this message can be one of the following:

- The indicated user ID does not exist for the BADSPOOLFILEID option in the configuration data set.
- The indicated user ID does not have proper authority.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Check that this user ID exists.

- If the user ID does not exist, correct the SMTP configuration data set, specifying a valid user ID, and restart SMTP.
- If the user ID does exist and RACF (or similar product) is used, ensure that the user ID has a security profile with create authority.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5167E Invalid Transfer To Userid: *userid*

Explanation: The indicated user ID does not exist for the RESTRICT TRANSFERTO statement in the configuration data set.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Check that this user ID exists. Otherwise, correct the SMTP configuration data set by specifying a valid user ID. Correct any network errors, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5168E Invalid Mailer Option: *option*

Explanation: The indicated option for the MAILER option in the configuration data set is not valid. See *z/OS Communications Server: IP Configuration Reference* for valid choices for the MAILER option.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5169E Invalid MailFileDSPrefix Option: *option*

Explanation: When editing the MAILFILEDSPREFIX statement in *hlq*.SMTP.CONFIG, the name can have a maximum of 20 characters. The characters must be either alphanumeric, or one of the following symbols: '.', '$', '#', or '@', and the name must meet data set name conventions. The default name for this prefix, used in place of the incorrect name, is the job that is running SMTP. For more information, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Correct the prefix name in the MAILFILEDSPREFIX statement to one that follows the guidelines specified above.

System Programmer Response: Assist the user as necessary.

Source Data Set: SMTP

Procedure Name: Error

EZA5170E Invalid MailFileVolume Option: *option*

Explanation: The MAILFILEVOLUME statement in *hlq*.SMTP.CONFIG has an incorrect parameter. The name of the volume cannot exceed 6 characters. The option is ignored. For more information on *hlq*.SMTP.CONFIG and the MAILFILEVOLUME, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Correct the name of the volume name in the MAILFILEVOLUME statement in *hlq*.SMTP.CONFIG to have 6 or fewer characters.

System Programmer Response: Assist the user as necessary.

Source Data Set: SMTP

Procedure Name: Error

EZA5171E Invalid MailFileUnit Option: *data set name*

Explanation: There are more than 8 characters specified in the name of the unit in the MAILFILEUNIT statement in *hlq*.SMTP.CONFIG file. For more information on the MAKEFILEUNIT statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Edit the MAILFILEUNIT statement in *hlq*.SMTP.CONFIG to make the name of the new unit fewer than or equal to 8 characters.

System Programmer Response: Assist the user as necessary.

Source Data Set: SMTP

Procedure Name: Error

EZA5172E Invalid TIMEZONE specified: *time zone*

Explanation: The TIMEZONE statement in *hlq*.SMTP.CONFIG contains an incorrect number of characters. The character string for the TIMEZONE statement can be 1–5 characters long. For more information on the TIMEZONE statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the TIMEZONE statement in *hlq*.SMTP.CONFIG to a printable character string that is 1–5 characters long.

System Programmer Response: Assist the user as necessary.

Source Data Set: SMTP

Procedure Name: Error

EZA5175I Mail Logging:

Explanation: The LOG statement is present in *hlq*.SMTP.CONFIG. SMTP is outputting the mail status to the data set named or pointed to by the LOGFILE DD statement in *hlq*.SEZAINST(SMTPPROC).

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5176E LOGFILE DD card missing, NOLOG forced.

Explanation: The LOG statement was present in *hlq*.SMTP.CONFIG and there was no LOGFILE DD parameter in *hlq*.SEZAINST(SMTPPROC) specifying the name of the data set used for the mail logging. SMTP does not log the mail.

System Action: SMTP continues.

User or Operator Response: Change the LOGFILE DD statement in the *hlq*.SEZAINST (SMTPROC) to a valid data set name. For more information on the LOGFILE DD statement and the configuring process, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5177E Invalid LocalClass Option: *class*

Explanation: The class specifies the spool class for local mail delivered by SMTP. The class specified in *hlq*.SMTP.CONFIG is more than 1 character long.

System Action: SMTP continues.

User or Operator Response: Change the LOCALCLASS statement in *hlq*.SMTP.CONFIG to a single character. This option is site-dependent. Ask the system administrator for site-dependent information.

System Programmer Response: Provide the user with site-dependent information as necessary. The default setting for the LOCALCLASS statement is B.

Source Data Set: SMTP

Procedure Name: Error

EZA5178E Invalid NJEClass Option: *class*

Explanation: More than 1 character was specified in the NJECLASS statement of *hlq*.SMTP.CONFIG. There can only be a single character specified. The NJECLASS statement specifies the spool class for mail delivered by SMTP to the NJE network. This option is site-dependent. Check with the system administrator for site-dependent information.

System Action: SMTP continues.

User or Operator Response: Change the NJECLASS option in *hlq*.SMTP.CONFIG to a single character specified by the system administrator. For more information on the NJECLASS statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Provide the user with site-dependent information as necessary. The default setting for the LOCALCLASS statement is B.

Source Data Set: SMTP

Procedure Name: Error

EZA5179I Debugging Enabled: *yes|no*

Explanation: Causes SMTP to record SMTP commands and replies in the data set specified in the SYSDEBUG DD statement. The SMTP connection number is recorded along with each SMTP command or reply. To change the data set pointed to, change the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). For more information on the DEBUG statement and the SYSDEBUG DD statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5180E SYSDEBUG DD card missing, DEBUG option ignored.

Explanation: The DEBUG statement is present in *hlq*.SMTP.CONFIG, but there is no SYSDEBUG DD statement in *hlq*.SEZAINST(SMTPPROC). Debug is not enabled.

System Action: SMTP continues.

User or Operator Response: Put a SYSDEBUG DD statement in the *hlq*.SEZAINST(SMTPPROC) specifying the data set name for SMTP's replies to the DEBUG commands. For more information on the DEBUG statement, the SYSDEBUG DD statement, and custom configuring, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5181E Invalid SpoolPollInterval: *seconds*

Explanation: The SPOOLPOLLINTERVAL statement has been incorrectly set. The appropriate values for the parameter of this statement are between 5 and 3600. The SPOOLPOLLINTERVAL statement specifies the interval for SMTP to check the spool for incoming spool files.

System Action: SMTP continues.

User or Operator Response: Change the parameter for the SPOOLPOLLINTERVAL statement in *hlq*.SMTP.CONFIG to a number between 5 and 3600. For more information on the SPOOLPOLLINTERVAL statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5183I Spool Poll Interval : *seconds* **seconds**

Explanation: Displays the interval for SMTP to check the spool for incoming spool files. To change this setting, you must change the SPOOLPOLLINTERVAL statement in the *hlq*.SMTP.CONFIG data set. For more information on the SPOOLPOLLINTERVAL statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5185I Mail from restricted users : Purge / Return / Transfer To *id*

Explanation: Informs the user that they have received mail from a user not allowed to utilize the SMTP services. The mail is not delivered but action taken on the mail is displayed. To make a user unrestricted, delete the RESTRICT statement containing the userID from *hlq*.SMTP.CONFIG. For more information on the RESTRICT statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5186I Mail File Dataset Prefix : *prefix*

Explanation: Displays the prefix that is added to the SMTP mail data sets. Data sets with this prefix contain mail that is in the process of being received or delivered. All data sets are cataloged. To change the prefix, change the parameter of the MAILFILEPREFIX statement in the *hlq*.SMTP.CONFIG data set. The prefix can be up to 20 characters in length.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5187I Mail File Unit Name : *unit name*

Explanation: Displays the unit where newly created SMTP mail data sets reside. To change this data set name, change the MAILFILEUNIT statement in the *hlq*.SMTP.CONFIG data set. Remember to use the data set naming guidelines in the *z/OS Communications Server: IP User's Guide and Commands*. For more information on the MAILFILEUNIT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5188I Mail File Volume : *volume*

Explanation: Displays the volume that newly allocated SMTP mail data set reside. If the name of the volume is not specified, SMTP allocates a storage volume. If your system does not have storage volumes, you must specify a volume name in the *hlq*.SMTP.CONFIG data set. For more information on the MAILFILEVOLUME statement and the *hlq*.SMTP.CONFIG data set, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5189I Local Output Class : *class*

Explanation: Displays the spool class for local mail delivered by SMTP. This option is site-dependent. Change the LOCALCLASS statement in the *hlq*.SMTP.CONFIG to change this setting. Ask the system administrator before setting to verify all site-dependent information. For more information on the LOCALCLASS statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Provide user with all site-dependent information.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5190I NJE Output Class : *class*

Explanation: Displays the spool class for mail delivered by SMTP to a NJE network. This option is site-dependent. To change this setting, change the NJECLASS statement in *hlq*.SMTP.CONFIG. Contact the system administrator before setting all site -dependent information. For more information on the NJECLASS statement and the *hlq*.SMTP.CONFIG data set, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5195I Unknown Command: *command*

Explanation: The indicated incorrect command was received on the server. The following are the valid commands:

- DATA
- EXPN
- HELP
- HELO
- MAIL
- NOOP
- QUEU
- QUIT
- RCPT
- RSET
- SAML
- SEND
- SOML
- TICK
- TURN
- VERB
- VRFY

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: The sender should correct the syntax before resubmitting the command.

Source Data Set: SMTPCMDS

Procedure Name: NoSuchCommand

EZA5196I *connection_number command_line*

Explanation: This message is written to the debug data set. The DEBUG statement is present in *hlq*.SMTP.CONFIG and SMTP is writing the command entered by the user to the data set specified by SYSDEBUG DD in *hlq*.SEZAINST(SMTPPROC). The DEBUG statement causes SMTP to record all commands and replies to a data set. To turn the DEBUG statement on or off, simply include or exclude the statement in *hlq*.SMTP.CONFIG respectively. For more information on the DEBUG statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPCMDS

Procedure Name: DoCommand

EZA5197I *connection_number* **Fragment:** *command*

Explanation: The DEBUG statement is present in the *hlq*.SMTP.CONFIG data set and this message indicates that SMTP is writing the command to the data set specified in *hlq*.SEZAINST(SMTPPROC). by the SYSDEBUG statement. The DEBUG statement causes SMTP to record SMTP commands and replies to a data set. For more information on the DEBUG statement and the configuring data sets, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPCMDS

Procedure Name: PrintResQueue

EZA5198I *connection_number reply*

Explanation: This message is written to the debug data set specified in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). To turn the debug option on or off, simply include or exclude the DEBUG statement in *hlq*.SMTP.CONFIG, respectively. For more information on the DEBUG statement, see the *z/OS Communications Server: IP Configuration Reference* .

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRPLY

Procedure Name: DoReply

EZA5199I *connection_number* **Fragment:** *reply*

Explanation: This message is written to the debug data set specified in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). To turn the debug option on or off, simply include or exclude the DEBUG statement in *hlq*.SMTP.CONFIG, respectively. For more information on the DEBUG statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRPLY

Procedure Name: DoReply

EZA5200I *connection_number reply*

Explanation: This message is written to the debug data set specified in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). To turn the debug option on or off, simply include or exclude the DEBUG statement in *hlq*.SMTP.CONFIG, respectively. For more information on the DEBUG statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response:

System Programmer Response: None.

Source Data Set: SMTPUTIL

Procedure Name: SendReply

EZA5201I *date time* **Looping Mail Transferred to** *userid*

Explanation: SMTP sends this message to the console error log to show that SMTP encountered a looping mail condition. This condition exists because the number of mailing attempts has exceeded the looping mail threshold limit. The mail in question is transferred to the indicated user ID.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Examine the piece of looping mail to see which user ID is causing the error. Because this error frequently occurs between the SMTP server and a MAILER, examine both for configuration errors.

Source Data Set: SMTPBTCH

Procedure Name: MakeBatchSMTPErrorFile

EZA5201I *date time* **Rcpt to unknown host:** *host*

Explanation: An RCPT TO: command was issued to the indicated host machine, and the SMTP server received a negative reply from a Domain Name Server. If the SMTP server is not using a Domain Name Server, the server could not find the indicated host name in the site tables.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: If the host name exists, check the Domain Name Server or site tables for configuration errors, and resend the mail message.

Source Data Set: SMTPRES

Procedure Name: ProcessRQR

EZA5201I *date time* **Rejecting command line too long**

Explanation: The argument string supplied after a command keyword exceeds the maximum allowable argument string length of 512 characters.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: If the error persists, determine which host is sending the incorrect command line, and correct the SMTP server.

Source Data Set: SMTPCMDS

Procedure Name: DoCommand

EZA5201I *date time* **Syntax Error: args after** *command* **command:** *command_line*

Explanation: SMTP sends this message to the server console when SMTP receives an incorrect DATA, QUIT, RSET, NOOP, TURN, or QUEU command. This message shows that the arguments with these commands are incorrect and unnecessary. The incorrect argument string is indicated at the end of the message.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: The sender should either correct the syntax before resubmitting the command, or issue the HELP command.

Source Data Set: SMTPCMDS

Procedure Name: DoCommand

EZA5209I Local Time Zone : *time zone*

Explanation: Displays the local time zone. To change the time zone, change the TIMEZONE statement in *hlq*.SMTP.CONFIG. The time zone must be exactly 3 characters. The default time zone is LCL. For more information on the TIMEZONE statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5210E Bad format SMTPNJE.HOSTINFO file found. Please run SMTPNJE

EZA5211E to create a new SMTPNJE.HOSTINFO file.

Explanation: The format must be updated, so run SMTPNJE.

System Action: SMTP continues.

User or Operator Response: Run the SMTPNJE program.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: AmpXAloc

EZA5212I *connection_number* **Data arrived while Sending Mail Body:**

Explanation: The DEBUG statement is present in *hlq*.SMTP.CONFIG and SMTP is writing this reply to the data set pointed to or named in *hlq*.SEZAINST(SMTPPROC) under the SYSDEBUG DD statement. For more information on the DEBUG statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: To view the commands and the responses, view the data set specified or pointed to by *hlq*.SEZAINST(SMTPPROC).

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5214I *character*

Explanation: Writes the data that arrived while SMTP was sending the mail body to the data set specified or pointed to in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). For more information on the DEBUG statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: To view the commands and the responses, view the data set specified or pointed to by *hlq*.SEZAINST(SMTPPROC).

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5216I *connection_number*: **Connection Inactive - Timed Out**

Explanation: SMTP has closed the connection because the time allotted for inactivity has passed. This message is written to the data set specified in the SYSDEBUG DD statement in *hlq*.SEZAINST(SMTPPROC). To change the amount of time given before SMTP closes the connection, change the INACTIVE statement in *hlq*.SMTP.CONFIG. For more information on the INACTIVE statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP closes the connection.

User or Operator Response: To use SMTP again, restart the program.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5217E Invalid numeric operand found: *number*

Explanation: The number entered was either too large or contained incorrect numbers. The numeric operand cannot have more than 10 digits and it cannot exceed 2 147 483 647. Also, there can be nothing but digits between 0 and 9 composing the operand.

System Action: SMTP continues.

User or Operator Response: Verify that the number entered follows the above guidelines.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5218E Invalid ServerPort specified: *port*

Explanation: The port number specified in the PORT statement of the *hlq*.PROFILE.TCPIP and the *hlq*.SMTP.CONFIG data sets must be between 0 and 65 535 and they must both be of equal value. The default port number for both configuration data sets is 25.

System Action: SMTP continues.

User or Operator Response: Change the PORT statement in the *hlq*.PROFILE.TCPIP or the *hlq*.SMTP.CONFIG data sets to be of equal value. For more information on the PORT statement, the *hlq*.PROFILE.TCPIP, and the *hlq*.SMTP.CONFIG data sets see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5220I *connection number*: **Ignoring Data/Commands received after QUIT**

Explanation: The quit command has been entered, and SMTP is in the process of closing the connection, so the commands issued after it are ignored.

System Action: SMTP continues closing the connection.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5221I NonExistent: *rc*

Explanation: The connection no longer exists for the indicated return code reason. The connection could have been dropped because of a software error on the remote SMTP server or because of a routing or connectivity problem in the internet network.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Normally, no action is necessary. The mail is delivered automatically at a later time when connectivity is restored. If the error persists, check for a software error in the remote SMTP server or for an error in the piece of mail itself. For example, if a user tries to send a binary piece of SMTP mail, some hosts' SMTP servers malfunction or abnormally close the connection.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5222I StoreData: entering "discard" mode

Explanation: SMTP sends the message to the local console to show that the sender's mail data set exceeded the maximum length specified by the MAXMAILBYTES option in the *hlq*.PROFILE.TCPIP data set. The excessively large mail data set is erased. This message is issued before the message that confirms the erasing of the data set.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Tell the sender to shorten the mail data set or to subdivide the data set into shorter data sets that are within the existing MAXMAILBYTES limitation. Increase the MAXMAILBYTES option value in the SMTP.CONFIG configuration data set, and tell the sender to try again.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5223I StoreData: leaving "discard" mode

Explanation: SMTP sends this message to the local console to show that the sender's mail data set exceeded the maximum length specified by the MAXMAILBYTES option in the *hlq*.PROFILE.TCPIP data set. This message also verifies that an attempt was made to return a message to the mail sender to inform the sender of a configuration problem on the receiving machine. The mail data set being processed is discarded.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Tell the sender to shorten the mail data set or to subdivide the data set into shorter data sets that are within the existing MAXMAILBYTES limitation. Increase the MAXMAILBYTES option value in the SMTP.CONFIG configuration data set, and tell the sender to try again.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5225I Resolver tracing : Enabled/Disabled

Explanation: Displays the status of the resolver tracer. The tracer causes a complete trace of all queries to and responses from the name server to be written on the user's console. To toggle between Enabled and Disabled, use the TRACE RESOLVER statement in the TCPIP.DATA file. The TRACE RESOLVER statement is used for debugging purposes only. For more information on the TRACE RESOLVER statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5228I Name Resolution Method *method*

Explanation: This message displays the method that SMTP uses for name resolution The method selected is dependent on the value associated with the RESOLVERUSAGE statement in the SMTP configuration file and whether NSINTERADDR statement is coded in the TCPIP.DATA file for a name server.

method is one of the following:

Host Tables
Indicates that RESOLVERUSAGE is YES and no name server is configured in the TCPIP.DATA file

Nameserver, Port *port_number*
Indicates that RESOLVERUSAGE is YES and one or more name servers are configured in the TCPIP.DATA file. *port_number* is the port number of the name server.

IPMAILERADDRESS or MAILER statement
Indicates that RESOLVERUSAGE is NO. The mail is directed to either to the IP address associated with the IPMAILERADDRESS statement or to the mailer associated with the MAILER statement in the SMTP configuration file.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5231I Nameserver Address *(port number)***:** *address*

Explanation: Displays the name server's address and its corresponding port number.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5232I Nameserver Response Timeout : *seconds* **seconds.**

Explanation: Displays the amount of time that is allotted for the name server to respond before the resolver closes the connection. To change the number displayed, change the RESOLVERTIMEOUT statement in TCPIP.DATA. For more information on the RESOLVERTIMEOUT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5233I Resolver Retry Interval : *minutes* **minutes**

Explanation: Displays the number of minutes SMTP waits between attempts to resolve domain names. To change this value, change the RESOLVERRETRYINT statement in *hlq*.SMTP.CONFIG. For more information on the RESOLVERRETRYINT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5234I UDP Retries/Nameserver/Interval: *seconds*

Explanation: To change this value, change the UDPRETRYINTERVAL command in *hlq*.NSMAIN.DATA. The UDPRETRYINTERVAL command specifies the amount of time the name server should wait before sending a query to another name server. The default for this command is 5 seconds. For more information on the UDPRETRYINTERVAL command, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5235I RCPT TO: response delay : *seconds* **seconds**

Explanation: Displays how long the SMTP server delays responding to the RCPT commands from the sender SMTP, while it is waiting for domain name resolution. If resolution does not occur before the specified time, the server assumes the name resolution is successful and queues the recipient address for asynchronous resolution. SMTP also sends the following message to the sender SMTP: 250 ok. The mail is returned to the sender if SMTP later determines that the recipient address cannot be resolved. To change this delay, change the RCPTRESPONSEDELAY statement in SMTP.SMTP.CONFIG. For more information on the RCPTRESPONSEDELAY statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5236I Max Temporary Error Retries: *Disabled* **/** *number*

Explanation: Displays the maximum number of times SMTP tries to redeliver mail to a host with a temporary problem. If the number of retries is 0, "*Disabled*" is displayed. Otherwise, the value for the TEMPERRORRETRIES statement is displayed. To change this value, change the TEMPERRORRETRIES statement in *hlq*.SMTP.CONFIG. For more information on the TEMPERRORRETRIES statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5240E Option Conflict: SECURE and RESTRICT may not both be specified

Explanation: The options SECURE and RESTRICT cannot be used together because they are incompatible.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5241E Invalid DBCS Option: *option*

Explanation: An invalid option has been specified in the *hlq*.SMTP.CONFIG data set for the DBCS statement. DBCS conversion is disabled.

System Action: SMTP continues.

User or Operator Response: Specify a valid option for the DBCS statement and restart SMTP. See *z/OS Communications Server: IP Configuration Reference* for more information about configuring DBCS support for the SMTP server.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5242E Invalid Kanji shift-in Option: *option*

Explanation: An invalid shift-in option has been specified in the *hlq*.SMTP.CONFIG data set for the DBCS statement. The valid shift-in options for the JIS78KJ and JIS83KJ conversion types are ASCII and JISROMAN. DBCS conversion is disabled.

System Action: SMTP continues.

User or Operator Response: Specify a valid shift-in option for the DBCS statement and restart SMTP. See *z/OS Communications Server: IP Configuration Reference* for more information about configuring DBCS support for the SMTP server.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5243E Too Many ALTTCPHOSTNAME statements - limit is *number*

Explanation: The number of alternative TCP host name options selected exceeds the maximum number allowed. The maximum number allowed is indicated at the end of the message; the default is 16.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Reduce the number of ALTTCPHOSTNAME statements in the configuration data set, or increase the MaxAltOurNames parameter to the desired number. Restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5256I Alternate TCP Domain Name*number* **:** *domain name*

Explanation: Displays the alternate, fully qualified host name by which SMTP recognizes the local host. To change the alternate domain name, change the ALTTCPHOSTNAME statement in *hlq*.SMTP.CONFIG. There can be up to 16 alternate host names. For more information on the ALTTCPHOSTNAME statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5257I Alternate TCP Domain Name*number* **:** *domain name*

Explanation: Displays the alternate, fully qualified host name by which SMTP recognizes the local host. To change the alternate domain name, change the ALTTCPHOSTNAME statement in *hlq*.SMTP.CONFIG. There can be up to 16 alternate host names. For more information on the ALTTCPHOSTNAME statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5258I Protocol Error. I= / Bytes Received = *bytes*

Explanation: This message is sent to the local console to show that the sender's mail was not successfully delivered because of a protocol Recoverable error. The SMTP server received the string, <CR><LF>·<CR><LF>, followed by additional data. The string, <CR><LF>·<CR><LF>, is used to specify the end of the data; thus, the client SMTP server is in error. The data is accepted, and the client SMTP server waits for an additional <CR><LF>·<CR><LF> string.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Describe the error to the postmaster of the SMTP server that is sending the incorrectly formatted mail.

Source Data Set: SMTPEVNT

Procedure Name: StoreDataTableErrors

EZA5259I Reply Line too long

Explanation: SMTP sends this message to the server console to show that a remote SMTP server's response exceeded the 512-byte limitation specified by RFC 821. SMTP accepts the reply and truncates it to 512 bytes. See Appendix C, "Related protocol specifications (RFCs)" on page 587 for information about accessing RFCs.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: If the error persists, notify the postmaster of the remote SMTP server the SMTP server is sending replies that are greater than the 512-byte limit.

Source Data Set: SMTPRPLY

Procedure Name: DoReply

EZA5260I Rewrite 822 Mail Headers : Enabled using rules from *data set*/Enabled using default rules/Disabled

Explanation: Displays the status of the REWRITE822HEADER statement in *hlq*.SMTP.CONFIG. The REWRITE822HEADER statement specifies whether SMTP should rewrite or print the RFC 822 headers of mail arriving from the NJE side of the mail gateway. There are 3 possible combinations of parameters in this statement. For a description of these combinations, and more information on the REWRITE822HEADER statement, see the *z/OS Communications Server: IP Configuration Reference*. See Appendix C, "Related protocol specifications (RFCs)" on page 587 for information about accessing RFCs.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5263I DBCS Conversion : *conversion_status*

Explanation: This message indicates the DBCS conversion status for the SMTP server at initialization time. The status indicates the DBCS conversion type that will be used when transferring mail, or "disabled," if DBCS conversion is not configured, or if the required DBCS translation table could not be loaded. DBCS conversion is either enabled, or disabled, according to the displayed status. For a detailed explanation of the DBCS conversion_status, see messages EZA5446N–EZA5456N.

System Action: SMTP continues.

User or Operator Response: If the status is "disabled," and a DBCS translation table could not be loaded, then 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 more information about the loading and customizing of DBCS translation tables.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5264I Alternate RSCS Domain Name : *domain name*

Explanation: Displays the alternate domain name of the NJE network when SMTP is running as a mail gateway. It can be specified only once. To change this value, change the ALTNJEDOMAIN statement in *hlq*.SMTP.CONFIG. For more information on the ALTNJEDOMAIN statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5265I Warn about mail older than : *days* **days**

Explanation: Displays how many days until SMTP returns mail to the sender with a header indicating the mail has been undeliverable for WARNINGAGE days. To change this value, change the WARNINGAGE statement in *hlq*.SMTP.CONFIG. For more information on the WARNINGAGE statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5272I Deliver via: *punch/netdata SR/NoSR* **Mailer** *temp: user ID@node ID local/NJE/unknown* **Addresses**

Explanation: Displays the format, source route status, the mailer state, and the addresses. To change any of these, consult the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5284E Invalid value *seconds* **specified for Inactive. Default used.**

Explanation: There is an incorrect value for the INACTIVE statement in *hlq*.SMTP.CONFIG. The INACTIVE statement specifies the number of seconds of inactivity after which SMTP considers the connection to be dead. The connection is then closed. The correct range for the INACTIVE statement is between 0 and 86 400® seconds. SMTP uses 180 for the INACTIVE statement's value. For more information on the INACTIVE statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the INACTIVE statement in *hlq*.SMTP.CONFIG to a correct value.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5285E Invalid value *seconds* **specified for FinishOpen. Default used.**

Explanation: There is an incorrect value in the FINISHOPEN statement in *hlq*.SMTP.CONFIG. The value must be between 0 and 86 400 seconds. SMPT uses 120 seconds as the value for FINISHOPEN. This statement specifies the number of seconds SMTP waits while trying to establish a connection to a remote site. After the time has passed, the connection is ended by SMTP. For more information on the FINISHOPEN statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the FINISHOPEN statement in *hlq*.SMTP.CONFIG to a correct value. If the default value is acceptable, remove the FINISHOPEN statement.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5286E Invalid value *days* **specified for RetryAge. Default used.**

Explanation: An incorrect value for the RETRYAGE statement in *hlq*.SMTP.CONFIG was used. The value must be between 0 and 365 days. The RETRYAGE statement specifies the number of days after which SMTP returns mail as undeliverable. SMTP uses 3 days as the value for the RETRYAGE statement. For more information on the RETRYAGE statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the parameter in the RETRYAGE statement in *hlq*.SMTP.CONFIG to a correct value.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5287E Invalid value *minutes* specified for RetryInt. Default used.

Explanation: The value of the RETRYINT statement must be between 0 and 1440 minutes. The RETRYINT statement specifies the number of minutes SMTP should wait between attempts to deliver the mail to a TCP host that is down. SMPT uses 20 minutes for the value of the RETRYINT statement. For more information on the RETRYINT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the RETRYINT statement in *hlq*.SMTP.CONFIG to a correct value.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5288E Invalid value *minutes* specified for ResolverRetryInt. Default used.

Explanation: An incorrect value was used for the RESOLVERRETRYINT statement in *hlq*.SMTP.CONFIG. This statement specifies the number of minutes SMTP waits between attempts to resolve domain names. The value for the RESOLVERRETRYINT statement must be between 0 and 1440 minutes. SMTP uses 20 minutes for the value of the RESOLVERRETRYINT statement. For more information on the RESOLVERRETRYINT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the RESOLVERRETRYINT statement in *hlq*.SMTP.CONFIG.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5289E Invalid value *value* specified for RcptResponseDelay. Default used.

Explanation: The error is caused either by an incorrect value of less than 0 or greater than 86 400, or because NSInterAddr is not coded in the TCPIP.DATA data set. SMTP uses a default of 60.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the SMTP configuration data set or the TCPIP.DATA data set.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5290E Invalid value *seconds* specified for TempErrorRetries. Default used.

Explanation: The value of the TEMPERRORRETRIES statement in *hlq*.SMTP.CONFIG must be either 0 or positive. This statement specifies the number of times SMTP tries to redeliver mail to a host with a temporary problem. SMTP uses the default value for the TEMPERRORRETRIES statement.

System Action: SMTP continues.

User or Operator Response: Correct the TEMPERRORRETRIES statement only when mail servers repeatedly terminate abnormally or hang SMTP mail transactions. If a change is necessary, change the TEMPERRORRETRIES statement in *hlq*.SMTP.CONFIG. For more information on the TEMPERRORRETRIES statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5294E Invalid value *value* specified for Inactive.

Explanation: The value for the INACTIVE statement was less than 0. This statement specifies the number of seconds of inactivity after which SMTP considers the connection dead. After the specified amount of inactive time, SMTP closes the connection. SMTP continues using the default value for the INACTIVE statement.

System Action: SMTP continues.

User or Operator Response: Change the INACTIVE statement in *hlq*.SMTP.CONFIG to a valid positive number. For more information on the INACTIVE statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5295E Invalid value *seconds* specified for FinishOpen.

Explanation: The FINISHOPEN statement specifies the number of seconds SMTP waits while trying to establish a connection to a remote site. After the specified number of seconds, SMTP ends the connection. The parameter for this statement must be between 1 and 86 400 seconds. The default value for this statement is 120 seconds. For more information on the FINISHOPEN statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the FINISHOPEN statement in *hlq*.SMTP.CONFIG.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5296E Invalid value *number* for RetryAge.

Explanation: The RETRYAGE statement in *hlq*.SMTP.CONFIG must be positive. This statement specifies the number of days after which SMTP returns mil as undeliverable. SMTP uses the default value for the RETRYAGE statement, which is 3 days. For more information on the RETRYAGE statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the RETRYAGE statement in *hlq*.SMTP.CONFIG to a correct value.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5297E Invalid value *minutes* specified for RetryInt.

Explanation: The RETRYINT statement in *hlq*.SMTP.CONFIG must contain a positive value. This statement specifies the number of minutes SMTP should wait between attempts to deliver mail to a TCP host that is down. The range of correct values for the RETRYINT statement is 0 and 1440 minutes. SMTP uses the default for the RETRYINT statement, which is 20 minutes. For more information on the RETRYINT statement and *hlq*.SMTP.CONFIG, see the *z/OS Communications Server: IP Configuration Reference*

System Action: SMTP continues.

User or Operator Response: Change the RETRYINT statement in *hlq*.SMTP.CONFIG to a correct value.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5298E Invalid value *minutes* specified for ResolverRetryInt.

Explanation: The value for the RESOLVERRETRYINT statement must be a positive number. This statement specifies the number of minutes SMTP waits between attempts to resolve domain names. SMTP uses the default value for the RESOLVERRETRYINT statement, which is 20 minutes. For more information on the RESOLVERRETRYINT statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the RESOLVERRETRYINT statement in *hlq*.SMTP.CONFIG to a positive, correct value.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5299E Invalid value *number* specified for RcptResponseDelay

Explanation: The value for the RCPTRESPONSEDELAY statement must be positive. This statement specifies how long the SMTP server delays responding to the RCPT commands from the sender SMTP. While it is waiting for domain name resolution. If resolution does not complete before the specified period, the SMTP server assumes name resolution is successful and sends a reply code of 250 ok. to the sender SMTP and queues the recipient address for asynchronous resolution. If the name resolution is later determined to be unresolved, the mail is returned to the sender. SMTP uses the default value for the RCPTRESPONSEDELAY statement, which is 60 seconds. For more information on this statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the parameter of the statement to a correct, positive value in *hlq*.SMTP.CONFIG.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5300E Invalid value *number* specified for TempErrorRetries.

Explanation: The value for the TEMPERRORRETRIES in *hlq*.SMTP.CONFIG must be a positive number. SMTP uses the default settings for the TEMPERRORRETRIES statement. For more information on the TEMPERRORRTTRIES statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Enter a correct, positive number in the TEMPERRORRTTRIES statement in *hlq*.SMTP.CONFIG.

System Programmer Response: Assist the user as necessary.

Source Data Set: SMTP

Procedure Name: Error

EZA5301E Invalid value *days* specified for WarningAge.

Explanation: An incorrect value was specified for the WARNINGAGE statement. SMTP uses the default setting. For more information on the WARNINGAGE statement and *hlq*.SMTP.CONFIG see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: Change the WARNINGAGE statement in *hlq*.SMTP.CONFIG to a valid number.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5302E OLD Option of Mailer statement no longer supported.

Explanation: This message only occurs when TCPIP parses the *hlq*.SMTP.CONFIG data set. The OLD parameter to the MAILER statement is no longer valid. If you are updating from an earlier release of TCP/IP, be aware that in Version 3 it is an error to specify OLD in the MAILER statement.

System Action: The SMTP server halts until the error is corrected.

User or Operator Response: None.

System Programmer Response: Remove the OLD parameter from the MAILER statement in your *hlq*.SMTP.CONFIG data set and reinitialize SMTP.

Source Data Set: SMTP

Procedure Name: Error

EZA5303E Invalid Address in SMSG Authorization List: *address*

Explanation: A syntax error has been found in the address of the local user ID you entered as a parameter to the SMSGAUTHLIST statement in your *hlq*.SMTP.CONFIG data set.

System Action: The SMTP server halts until the error is corrected.

User or Operator Response: None.

System Programmer Response: Correct the address that you entered in the SMSGAUTHLIST statement and reinitialize SMTP.

Source Data Set: SMTP

Procedure Name: Error

EZA5311E LOOPINGMAIL statement no longer valid. Please use the BADSPOOLFILEID statement

Explanation: Earlier versions of TCP/IP used the LOOPINGMAIL statement to specify a user ID on the local system where SMTP transfers unreadable spool files and looping mail. This statement has been replaced by the BADSPOOLFILEID statement in Version 3.

System Action: The SMTP server is halted until the statement is corrected.

User or Operator Response: None.

System Programmer Response: Replace the LOOPINGMAIL statement with an equivalent BADSPOOLFILEID statement in your *hlq*.SMTP.CONFIG data set and reinitialize SMTP.

Source Data Set: SMTP

Procedure Name: Error

EZA5312E Only one POSTMASTER statement may be specified in SECURE mode

Explanation: If you have entered a SECURE statement in your *hlq*.SMTP.CONFIG data set, you can only use one POSTMASTER statement, and you can only specify a local user ID as the recipient of mail addressed to POSTMASTER.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: None.

System Programmer Response: Make sure that you list only one POSTMASTER statement in your *hlq*.SMTP.CONFIG data set. Also make sure that you specify a local user ID instead of a user address. Reinitialize SMTP.

Source Data Set: SMTP

Procedure Name: Error

EZA5313E Invalid POSTMASTER address: *address*

Explanation: The NJE or SMTP address in the POSTMASTER statement of your *hlq*.SMTP.CONFIG data set is incorrect.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: None.

System Programmer Response: Make sure that you have entered a correct, valid local user ID or a correct, valid NJE or SMTP address as part of your POSTMASTER statement. Reinitialize SMTP.

Source Data Set: SMTP

Procedure Name: Error

EZA5314E In SECURE mode, POSTMASTER address must be a valid local userid

Explanation: If you have entered a SECURE statement in your *hlq*.SMTP.CONFIG data set, then the POSTMASTER statement you use must take the form of a valid local user ID. You cannot enter a user address if you are in the SECURE mode.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: None.

System Programmer Response: Make sure to enter only one valid local user ID in your POSTMASTER statement, then reinitialize SMTP. See also message EZA5312I.

Source Data Set: SMTP

Procedure Name: Error

EZA5315E Expecting 'PRINT' or 'NOPRINT' after 'REWRITE822HEADER YES' statement

Explanation: If you entered a REWRITE822HEADER YES statement in your *hlq*.SMTP.CONFIG data set, you must also enter the PRINT parameter or the NOPRINT parameter for the RFC 822 headers. See Appendix C, "Related protocol specifications (RFCs)" on page 587 for information about accessing RFCs.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: After the REWRITE822HEADER YES statement, enter PRINT or NOPRINT, depending on whether you want the RFC 822 headers to display on your console. For more information, see *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: Error

EZA5317I Listen on Address : Unspecified

Explanation: The LISTENONADDRESS configuration statement of the *hlq*.SMTP.CONFIG data set was used without the IP address specified. The LISTENONADDRESS statement defines which HOME IP address to use for a mutlhome system.

System Action: SMTP continues.

User or Operator Response: Reissue the LISTENONADDRESS statement specifying a HOME IP address. For more information about the LISTENONADDRESS statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5318I Listen on Address : *IP address*

Explanation: The LISTENONADDRESS statement of the *hlq*.SMTP.CONFIG data set is parsed and the indicated IP address is confirmed. For more information about the LISTENONADDRESS statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5319I Outbound Open Session Limit : None

Explanation: The OUTBOUNDOPENLIMIT statement limits the number of simultaneous TCP connections in use for mail delivery. The default value is 0. For more information, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5320I Outbound Open Session Limit : *outbound open limit*

Explanation: If you have specified the number of connections as a parameter to the OUTBOUNDOPENLIMIT statement in your *hlq*.SMTP.CONFIG data set, this message displays the limit set. The OUTBOUNDOPENLIMIT statement should only be used if limited TCP resources exist on the system and SMTP is using an undue portion of those resources.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5321I SMSG Authorization List : None

Explanation: You have not authorized any local users to issue privileged SMTP SMSG commands. Only users specified in the SMSGAUTHLIST statement of the *hlq*.SMTP.CONFIG data set can issue the privileged commands. For more information about these commands, see *z/OS Communications Server: IP User's Guide and Commands*. See also message EZA5322I.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5322I SMSG Authorization List : Processed

Explanation: TCPIP has processed the user IDs you entered as parameters to the SMSGAUTHLIST statement in your *hlq*.SMTP.CONFIG data set. Those user IDs are now authorized to issue the privileged SMTP SMSG commands. For more information about the privileged commands, see *z/OS Communications Server: IP User's Guide and Commands*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5324I *procedure_name* **Terminating: (***return_code***)**

Explanation: SMTP has received a TCP/IP shut down return code from the TCP/IP stack; therefore, SMTP will also terminate.

procedure_name is the SMTP procedure name.

(*return_code*) is the TCP/IP return code. This return code will always be (8558) which indicates that TCP/IP shut down.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPUTIL

Procedure Name: CheckResult

EZA5325E Unable to delete existing file *FullDatasetName***, RC=***nn***, DSPrefix may be invalid. SMTP terminating.**

Explanation: SMTP attempts to allocate the data set specified in the error message to determine if SMTP has authority to delete data sets beginning with the defined MAILFILEDSPREFIX value. Because the data set already exists, SMTP attempts to delete it, and receives an error from the operating system.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: Ensure that SMTP has the proper Security Authorization Facility (SAF) authority to delete data sets beginning with the prefix defined by MAILFILEDSPREFIX. Check for any system error messages that appear concurrently with this error message.

If return code is 208, check to see if there is data set contention for the data set name displayed in the error message. SMTP must have access to all its data sets to perform its function. Therefore, if a defragmentation utility (or any other application) is actively accessing SMTP data sets while the SMTP server is active, a data set contention problem could occur. To avoid this problem, configure the application to bypass data sets with the SMTP high level qualifier. The high level qualifier is set by the SMTP configuration parameter MAILFILEDSPREFIX.

Source Data Set: SMTP

Procedure Name: InitializeDeliveryQueue

EZA5326E Unable to allocate new file *fulldatasetname***, RC=** *nn***, DSPrefix may be invalid. SMTP terminating.**

Explanation: SMTP attempts to allocate the data set specified in the error message to determine if SMTP has authority to delete data sets beginning with the defined MAILFILEDSPREFIX value. It receives an error from the operating system.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: Ensure that SMTP has the proper SAF authority to delete data sets beginning with the prefix defined by MAILFILEDSPREFIX. Check for any system error messages that appear concurrently with this error message.

Source Data Set: SMTP InitializeDeliveryQueue

EZA5327E Terminating: Cannot allocate *data set name*, **rc=** *return code*

Explanation: SMTP was unable to allocate a mail data set beginning with the prefix value specified in the MAILFILEDSPREFIX statement of your *hlq*.SMTP.CONFIG data set. An MVS operating system error was detected.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: Make sure that SMTP has the proper SAF authority to create or delete data sets beginning with the prefix defined in MAILFILEDSPREFIX. Check for any MVS system error messages that appear when this message does.

Source Data Set: SMTP

Procedure Name: InitializeDeliveryQueue

EZA5328E *date time* **Terminating: Stop received**

Explanation: SMTP received an external “Stop” interrupt from the operating system.

System Action: All files are closed and SMTP ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Check for any system error messages that appear concurrently with this error message.

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5329E *procedure name* **Terminating:** *errmsg (msgnum)*

Explanation: A condition has occurred that caused SMTP to end.

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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: SMTP ends.

User or Operator Response: Notify the system administrator.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: SMTPUTIL

Procedure Name: CheckResult

EZA5330I *procedure name* **SMTP TCP Returns:** *errmsg (msgnum)*

Explanation: This message indicates an error in the SMTP/TCP interface, as specified. It generally implies that the connection has been closed by the remote site. This message is generated only if the DEBUG statement is active in the SMTP startup procedure.

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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: SMTP continues.

User or Operator Response: Notify the system administrator.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: SMTPUTIL

Procedure Name: CheckResult

EZA5331E Local Error in MakeAddrBlok RC = *rc*

Explanation: SMTP did not have enough free disk space to store an ADDRBLOK data set.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: If the disk is full, delete old data sets, such as SMTP logs, or increase the size of the disk. If the disk is not full, there might be a problem with the ADDRBLOK data set. If the error persists, contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5332E Error occurred copying spool file, return code=*return code*

Explanation: An error occurred before the spool file finished copying. The spool file is closed.

System Action: Processing is terminated.

User or Operator Response: Use the return code value indicated to determine why the error is occuring and correct the problem or delete the file. Restart SMTP. For more information on return codes, refer to *z/OS Communications Server: IP Application Programming Interface Guide*.

System Programmer Response: None.

Source Data Set: SMTPBTCH

Procedure Name: InformUser

EZA5333E *data_set* **ADDRBLOK is in an inconsistent state...repairing**

Explanation: SMTP sends this message to the server console to show a problem with the ADDRBLOK data set. SMTP attempts to rewrite the ADDRBLOK data set in a correct format.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Normally, no response is necessary. If SMTP abends while performing this rewrite, the ADDRBLOK might not be recoverable. Rename or discard the ADDRBLOK data set to prevent further abends of SMTP.

Source Data Set: SMTPQUEU

Procedure Name: EnQueueForDelivery

EZA5334E Delivery of Note *note* **to** *address* **failed RC =** *rc*

Explanation: SMTP sends this message to the server console to show that the indicated note was not successfully transferred to the indicated address for the indicated return code reason. This condition occurs if SMTP tries to punch a file to an incorrect user ID, or if SMTP tries to punch a file with a header with more than 80 characters to a mailer running Mailer Old.

System Action: SMTP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Message EZA5336E preceeds this error message, so follow the information in EZA5336E. Return code = 53 indicates an incorrect user ID or user node. Determine why this was unsuccessful, correct the problem, and try again.

Source Data Set: SMTPQUEU

Procedure Name: DeliverSpoolFile

EZA5335E Unable to allocate new file *data set name* **rc=***return code*

Explanation: If SMTP cannot save the new data set to storage, this message provides a return code to explain the failure.

System Action: SMTP continues.

User or Operator Response: Retry the erase, close, or other I/O operation, making sure that the command parameters and syntax are correct. Use the return code information to correct the command.

System Programmer Response: None.

Source Data Set: SMMVSUB

Procedure Name: svc202

EZA5336E Error allocating spool, user=*user node.user ID***, class=***output class***, rc=***return code*

Explanation: This message indicates an error in delivering a mail file, either to a local user or to a user on the associated Job Entry Subsystem (JES). It specifies the user node and ID, the spool data set format, and the return code. The spool file is not delivered.

System Action: SMTP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: This problem usually occurs because the user ID is not a valid JES ID or the user node is not a valid NJE node for your system. Check that the user node is defined correctly in the IEFSSNxx parmlib member for your system. For more information about JES IDs, see the *z/OS JES3 Initialization and Tuning Guide*. If this is not the problem, use the return code value to determine why the error is occuring and correct the problem. Restart SMTP. For more information about spool return codes, see the *z/OS JES3 Messages* or *z/OS JES2 Messages*, depending on which version of JES you are using.

Note: Instead of using the iefssnxx.parmlib member to specify the JES node, you can use the keyword NJENODENAME within your SMTP configuration to specify a valid NJE node. For more information, see NJENODENAME in *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPQUEU

Procedure Name: DeliverSpoolFile

EZA5337E SMTP inactive, resource *full data set name* **not available rc=***return code*

Explanation: SMTP sends this message to the server console to show that SMTP cannot use one of its existing data sets.

System Action: SMTP continues.

User or Operator Response: Release the data set that SMTP is trying to use.

System Programmer Response: None.

Source Data Set: SMMVSUB

Procedure Name: svc202

EZA5338E SMTP inactive, resource *full data set name* **still not available rc=***return code*

Explanation: SMTP sends this message to the server console to show that SMTP still cannot use one of its existing data sets.

System Action: SMTP continues.

User or Operator Response: Release the data set that SMTP is trying to use.

System Programmer Response: None.

Source Data Set: SMMVSUB

Procedure Name: svc202

EZA5339E SMTP active, resource *full data set name* **now available rc=***return code*

Explanation: SMTP sends this message to the server console to show that SMTP now has the use of the data set for which it was waiting.

System Action: SMTP resumes normal operation.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMMVSUB

Procedure Name: svc202

EZA5340I SMTP NOTE from *userid* **at** *nodeid* **was truncated by** *hostname*

Explanation: The SMTP note is truncated because the note being read from the JES spool is too large and the CHECKSPOOLSIZE statement is enabled. The message is also appended to the truncated SMTP note. The truncated note is sent to its recipients. The MAXMAILBYTES statement in the *hlq*.SMTP.CONFIG file can be adjusted to change the primary space allocated for the *hlq*.TEMP.NOTE data set. For more information about the configuration statements see *z/OS Communications Server: IP Configuration Reference*.

userid is the user identifier that sent the SMTP note to the JES spool using TSO XMIT.

nodeid is the node identifier where the SMTP note originated.

hostname is the host where the SMTP mailer (which truncated the SMTP note) is active.

System Action: SMTP continues to run and large notes will be truncated.

User or Operator Response: Notify your system programmer about any truncated notes.

System Programmer Response: Review your system DASD resource requirements and adjust the MAXMAILBYTES statement, if appropriate.

Source Data Set: SMTPBTCH

Procedure Name: ServiceReaderMVS

EZA5341I CheckSpoolSize : Enabled

Explanation: The CheckSpoolSize statement in the *hlq*.SMTP.CONFIG file is enabled. If this statement is enabled, SMTP checks the size of the JES spool file and if it is larger than the primary space allocation for the data set *hlq*.TEMP.NOTE, then that SMTP note is truncated. The default for this statement is disabled. For more information about the CHECKSPOOLSIZE statement see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5342I SMTP NOTE from spool file job ID *jobid* was truncated by *hostname*

Explanation: The SMTP note is truncated because the note being read from JES spool is too large and the CHECKSPOOLSIZE statement is enabled. The message is also appended to the truncated SMTP note. The truncated note is sent to its recipients. The MAXMAILBYTES statement in the *hlq*.SMTP.CONFIG file can be adjusted to change the primary space allocated for the *hlq*.TEMP.NOTE data set. For more information about the configuration statements see *z/OS Communications Server: IP Configuration Reference*.

jobid is the spool file job identifier of the SMTP note used by SMTP batch.

hostname is the host where the SMTP mailer that truncated the SMTP note is active.

System Action: SMTP continues to run and large notes will be truncated.

User or Operator Response: Notify your system programmer about any truncated notes.

System Programmer Response: Review your system DASD resource requirements and adjust the MAXMAILBYTES statement if appropriate.

Source Data Set: SMTPBTCH

Procedure Name: ServiceReaderMVS

EZA5343I Translated ATSIGN symbol : *symbol*

Explanation: The ATSIGN statement in the *hlq*.SMTP.CONFIG file has been set. Setting this statement means that SMTP will use this character instead of the @ symbol between the user ID and host ID fields in addressing strings, for example, userid@hostid. This statement should only be used for specific languages because of translation encodings.

Setting this option might cause problems if notes are being sent between different language sites.

symbol is the input symbol replacing the @ symbol.

For more information about the ATSIGN configuration statement, refer to the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues

User or Operator Response: None

System Programmer Response: None

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5344E Incorrect ATSIGN symbol specified: *symbol*

Explanation: The input symbol replacing the @ symbol must be just one character in length. The specified input symbol is too large.

symbol is the input string replacing the @ symbol.

For more information about the ATSIGN configuration statement, refer to the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: Change the input string specified in the ATSIGN configuration statement to one character. Restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5345E Option conflict: the ATSIGN symbol and default REWRITE822 headers table cannot both be active

Explanation: The translate ATSIGN and REWRITE822 headers cannot both be activated because the default REWRITE822 header table contains a hardcoded @ symbol that cannot be modified. Therefore, these options cannot be used together.

For more information about the configuration statements, refer to the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP ends.

User or Operator Response: None.

System Programmer Response: Select one of the two options to activate in the *hlq*.SMTP.CONFIG file. Restart SMTP.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5346E Invalid IPMailer Address Specified: *IP address*

Explanation: The IP address specified in the IPMAILERADDRESS statement of your *hlq*.SMTP.CONFIG data set is not correct or not valid.

System Action: The SMTP server is halted until the problem is corrected.

User or Operator Response: None.

System Programmer Response: Verify the IP address of the batch SMTP server, and correct the IPMAILERADDRESS statement as necessary. For details, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTP

Procedure Name: Error

EZA5347E IPMAILERADDRESS statement conflicts with MAILER UNKNOWN statement

Explanation: The MAILER UNKNOWN statement in your *hlq*.SMTP.CONFIG data set is incompatible with the IPMAILERADDRESS statement.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: None.

System Programmer Response: Remove the UNKNOWN parameter from the MAILER statement. For details, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTP

Procedure Name: Error

EZA5348I IP Mailer Address : *IP address*

Explanation: This message specifies the current IP address for the IPMAILERADDRESS statement in the *hlq*.SMTP.CONFIG data set. If only the MAILER statement is used, the IP address is shown as “Disabled.”

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5351E Invalid NJENode Specified: *node name*

Explanation: An incorrect NJENODENAME statement has been specified in the *hlq*.SMTP.CONFIG data set.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: Refer this message to your system programmer.

System Programmer Response: Correct the NJENODENAME statement in the *hlq*.SMTP.CONFIG data set and restart SMTP. For more information, see the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTP

Procedure Name: Error

EZA5353E NJENode was specified after another option used its value. NJENode

---

EZA5354E must be specified prior to other configuration options.

Explanation: The NJENODENAME statement from the *hlq*.SMTP.CONFIG data set was parsed and a value that was previously defined for another statement was encountered. The NJENODENAME statement must be placed before the statement that is using the same value as that assigned to NJENODENAME.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: Refer this message to your system programmer.

System Programmer Response: Move the NJENODENAME statement above the statement using the same value as that assigned to NJENODENAME or change the value of either statement to avoid the conflict. Contact the IBM Software Support Center if more information is required.

Source Data Set: SMTP

Procedure Name: Error

EZA5355E GATEWAY specified but SMTPNJE DD card not found.

EZA5356E NJE gateway facility will not be enabled.

Explanation: The GATEWAY statement was included in the SMTP configuration data set but the input data set for SMTPNJE was not found. The NJE gateway facility is not enabled.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Issue the SMTPNJE command to create the input data set for SMTPNJE. Also, make sure that SMTPNJE is activated by removing the asterisk from the //SMTPNJE DD statement of the SMTP catalogued procedure. For more information, see ″Configuring the SMTP Server″ in the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTP

Procedure Name: AmpXAloc

EZA5364I MVS Debugging : Enabled

Explanation: The MVSDEBUG statement has been entered into the *hlq*.SMTP.CONFIG data set. This statement causes SMTP to do extensive debugging and should only be used when instructed by the IBM Software Support Center.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5378I Mailer : None

Explanation: If the MAILER statement in the *hlq*.SMTP.CONFIG data set specifies the NONJE parameter, then mail for recipients on the NJE network is spooled directly to the recipients. No batch SMTP server is defined for the associated NJE network.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5379E Invalid ListenOnAddress Specified: *IP address*

Explanation: The IP address specified for the LISTENONADDRESS statement in the *hlq*.SMTP.CONFIG data set is not valid. The address specified must match an existing HOME IP address as defined in the HOME statement of the *hlq*.TCPIP.PROFILE data set.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: Refer this message to your system programmer.

System Programmer Response: Correct the LISTENONADDRESS statement with a valid HOME IP address or add the IP address defined in the *hlq*.SMTP.CONFIG data set to the HOME statement of the *hlq*.TCPIP.PROFILE data set. For details, see *z/OS Communications Server: IP Configuration Reference* .

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5380E Using Default ListenOnAddress: *IP address*

Explanation: If the IP address parameter of the LISTENONADDRESS statement is not defined in the *hlq*.SMTP.CONFIG data set, this message indicates that SMTP is using the default value. For details see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Specify a valid, correct IP address for this statement as necessary.

Source Data Set: SMTP

Procedure Name: MarkForExpirationCheck

EZA5383E Syntax Error in POSTMASTER address:

EZA5384E *error message*

Explanation: An error exists in user ID or NJE address specified in the POSTMASTER statement of the *hlq*.SMTP.CONFIG data set, as indicated by the error message.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Enter the correct local user ID or NJE or SMTP address to which mail addressed to POSTMASTER should be delivered. For details, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPARSE

Procedure Name: ReplySyntaxError

EZA5385E Nickname: *nickname* **Address:** *address*

Explanation: SMTP was unable to parse the user ID and node ID you entered in the SMTP NAMES data set. SMTP could not validate the syntax of the resulting user address. The name is not added to the SMTP NAMES data set.

System Action: SMTP continues.

User or Operator Response: Verify the user ID and node ID for the intended recipient. For more information, see *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: SMTPNAME

Procedure Name: ValidateAddress

EZA5387E Ignoring duplicate definition of nickname: *nickname*

Explanation: A nickname already exists for the user ID and user node specified in your SMTP NAMES entry. SMTP ignores the entry.

System Action: SMTP continues.

User or Operator Response: Check your SMTP NAMES data set to verify the user ID and user node. For more information, see *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: SMTPNAME

Procedure Name: BuildNickNameTable

EZA5393I Rule *rule number***:**

Explanation: This message specifies the rule number. The *hlq*.SMTP.RULES data set contains the rewrite rules for header addresses in SMTP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRULE

Procedure Name: DisplayRulePrefix

EZA5397I *keyword name*

EZA5398I '*character string (non-case-sensitive)*'

EZA5399I "*character string (case-sensitive)*"

EZA5400I *character string (wildcard token)*

Explanation: This is the SMTP RULES keyword. When SMTP parses the rule in message EZA5393I, each string in the keyword is displayed along with one of these messages. As indicated, the specific message you see depends on whether the string is case-sensitive and whether it is a wildcard. Some keywords have special meaning to the rules interpreter. For more information, see *z/OS Communications Server: IP Configuration Reference*. See also message EZA5416E.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRULE

Procedure Name: EchoElem

EZA5401I FIELD ALIASES:

EZA5402I *alias name* **=**

EZA5403I **"***alias definition (case-sensitive)***"**

EZA5404I **'***alias definition (non-case-sensitive)***;'**

Explanation: These messages list the set of alias names (or field names) defined in the SMTP.RULES data set. As indicated, the specific message you see depends on whether the alias name is case-sensitive. For more information, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRULE

Procedure Name: EchoRules

EZA5406I REWRITE RULES:

EZA5407I **(***alias name***)**

Explanation: These messages specify the rewrite rules for the alias name shown in message EZA5407I. They precede the procedure EchoElem, whose output is explained in messages EZA5397I–EZA5400I. EchoElem parses the keywords for each rule for the alias name indicated.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRULE

Procedure Name: EchoRules

EZA5409I **=>** *NJE to TCPIP address transformation rule*

Explanation: This message specifies the rewrite rules for the alias name shown in message EZA5407I when the rule defines an address transformation from NJE to TCPIP. It precedes the procedure EchoElem, whose output is explained in messages EZA5397I–EZA5400I. EchoElem parses the keywords for each rule for the alias name indicated. See also message EZA5417I.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRULE

Procedure Name: EchoRules

EZA5412E *rule line*

EZA5414E SMTP RULES ERROR: *error message*

EZA5415E Line number *line number*

Explanation: The first of these 3 messages shows the line of code in the rule definition section of the SMTP.RULES data set where an error was found. The second message specifies the kind of error that was found, and the third message provides the line number where the error occurred.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: None.

System Programmer Response: Use the information provided by the error messages to correct the syntax of the rule definition. For more information, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPRULE

Procedure Name: PrintError

EZA5416E Error

Explanation: If the RULE keyword for the rule number specified in message EZA5393I contains a syntax error, this message appears when SMTP parses the SMTP.RULES data set.

System Action: The SMTP server is halted until the error is corrected.

User or Operator Response: None.

System Programmer Response: Correct the keyword syntax error. For details, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPRULE

Procedure Name: EchoElem

EZA5417I ->

Explanation: This message specifies the rewrite rules for the alias name shown in message EZA5407I when the rule defines an address transformation from TCPIP to NJE. It precedes the procedure EchoElem, whose output is explained in messages EZA5397I–EZA5400I. This procedure parses the keywords for each rule for the alias name indicated. See also message EZA5409I.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRULE

Procedure Name: EchoRules

EZA5423E Invalid Record in SMTP SECTABLE (the SMTP Security Table):

Explanation: SMTP encountered a syntax error while processing the security table. The next message output is a listing of the erroneous security table record. This message is followed by one or more other messages that describe specific syntax or logic problems with the record.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the errors in the security table data set, and restart SMTP.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5425E Expecting NodeId but found none.

Explanation: This message indicates the reason for error message EZA5423E. In this case the NJE node ID of the authorized user was omitted from the *hlq*.DD data set entry. The *hlq*.DD data set entry is not recognized.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Add the NJE node ID to the security table entry for the authorized user. For more information, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5426E Primary_*nickname/mbox* mapping must be 'Y' or 'N' but found: *nickname or mbox value*

Explanation: The only valid values for the primary nickname and primary mbox parameters in the *hlq*.DD data set are Y and N. You have specified an incorrect value.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Enter Y or N for the parameter indicated. For more information, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5430E Unexpected characters found after nickname

Explanation: The syntax of the *hlq*.DD data set record is incorrect. Extraneous characters or an excess number of characters have been entered after the nickname. See message EZA5426I.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the syntax of the security table as necessary. For more information, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5431E Duplicate Primary *nickname/mbox* **mapping detected.**

Explanation: Each nickname can have only one primary nickname record set to Y. Each NJE user ID, NJE node ID pair can have only one primary mailbox record set to Y. The duplicate primary nickname or primary mailbox record is ignored.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: See *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5435E Special characters not allowed in nickname

Explanation: The nickname field in the *hlq*.DD data set data set contains special characters. Only alphanumeric characters are allowed for nicknames.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Make sure that nicknames contain no special characters.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5439S Fatal Errors found in SMTP Security Table ... Halting!

Explanation: Either no primary mapping of the NJE node address is found or no primary mapping of the nickname is found. This message follows one or both of the error messages EZA5437E or EZA5438E.

System Action: The SMTP server halts until the error is corrected.

User or Operator Response: None.

System Programmer Response: Correct the *hlq*.DD data set security table, and restart SMTP.

Source Data Set: SMTPSEC

Procedure Name: CheckSecurityTable

EZA5440S Fatal Error: Unable to open SECTABLE DD but SECURE option was specified.

Explanation: The SECURE statement is used as a configuration statement in the *hlq*.SMTP.CONFIG data set, but no corresponding SECTABLE data set definition name is found in the SEZAINST(SMTPPROC) data set.

System Action: SMTP ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure that the SECTABLE DD statement name is included in the updated SMTP cataloged procedure *hlq*.SEZAINST(SMTPPROC). This points to the DD data set data set and allows SMTP to operate as a secure mail gateway between TCP network sites and NJE network sites. For more information, see *z/OS Communications Server: IP Configuration Reference*. Reinitialize SMTP.

Source Data Set: SMTPSEC

Procedure Name: CheckSecurityTable

EZA5441E Wildcard "*wildcard*" not allowed in

EZA5442I node name

EZA5443I nickname

EZA5444I field.

Explanation: The indicated wildcard token or string entered in either the node ID field or the nickname field of the record (see message EZA5423E for the specific security table record) is not a valid token or string for that field.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Correct the syntax of the DD data set record as indicated. See *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5445I NickName not allowed with wildcard userid field.

Explanation: If the NJE user ID is specified as a wildcard in the DD data set, then the nickname field must be left blank.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Delete the entry from the nicknames field, or change the user ID field in the DD data set to eliminate the wildcard. For details, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPSEC

Procedure Name: SecurityTableErrors

EZA5446N KANJI JIS 1978 shift-in ASCII

Explanation: The SMTP server and client programs access data sets containing data that is usually in extended binary coded decimal interchange code (EBCDIC) format. To transfer these data sets to or from an American National Standard Code for Information Interchange (ASCII) based host requires the use of a translation table. This message indicates that the SMTP client is configured to use the Japan Institute of Standards (JIS) 1978 version if kanji conversion is enabled. ASCII is output when the server converts from JIS kanji code. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5447N KANJI JIS 1978 shift-in JISROMAN

Explanation: The SMTP server and client programs access data sets containing data that is usually in extended binary coded decimal interchange code (EBCDIC) format. To transfer these data sets to or from an American National Standard Code for Information Interchange (ASCII) based host requires the use of a translation table. This message indicates that the SMTP client is configured to use the Japan Institute of Standards (JIS) 1978 version if kanji conversion is to be performed. JISROMAN is used when the server converts from JIS kanji code. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5448N KANJI JIS 1983 shift-in ASCII

Explanation: The SMTP client is configured to use the Japan Institute of Standards (JIS) 1983 version if kanji conversion is to be performed. It also indicates that ASCII is used when the SMTP server converts from JIS kanji code. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5449N KANJI JIS 1983 shift-in JISROMAN

Explanation: The SMTP client is configured to use the Japan Institute of Standards (JIS) 1983 version if kanji conversion is to be performed. It also indicates that JISROMAN is used when the server converts from JIS kanji code. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5450N SJISKANJI

Explanation: The SMTP client is configured to use the Japan Institute of Standards (JIS) version if the KANJI statement in the *hlq*.SMTP.CONFIG data set is enabled. It also indicates that the transfer type for the SMTP server is not changed. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5451N EUCKANJI

Explanation: The SMTP client is configured to use the Extended Unix Code kanji transfer type if the KANJI statement in the *hlq*.SMTP.CONFIG data set is enabled. It also indicates that the transfer type for the SMTP server is not changed. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5452N IBMKANJI

Explanation: The SMTP client is configured to use the IBM (EBCDIC) kanji transfer type if the KANJI statement in the *hlq*.SMTP.CONFIG data set is enabled. It also indicates that the transfer type for the SMTP server is not changed. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5453N HANGEUL

Explanation: The SMTP client is configured to use the Hangeul transfer type if the KANJI statement in the *hlq*.SMTP.CONFIG data set is enabled. It also indicates that the transfer type for the SMTP server is not changed. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5454N KSC-5601

Explanation: The SMTP client is configured to use the Korean Standard Code KSC-5601 if the KANJI statement in the *hlq*.SMTP.CONFIG data set is enabled. It also indicates that the transfer type for the SMTP server is not changed. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams -->

EZA5455N TRADITIONAL CHINESE

Explanation: The SMTP client is configured to use Traditional Chinese (5550) if the KANJI statement in the *hlq*.SMTP.CONFIG data set is enabled. It also indicates that the transfer type for the SMTP server is not changed. See also message EZA5447I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5456N Disabled. Unable to load: *DBCS standard file name.load file type*

Explanation: The double-byte character set (DBCS) translation table that you specified cannot be loaded. It is not supported by SMTP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: See *z/OS Communications Server: IP Configuration Reference* for the most recent list of DBCS translation data sets supported by SMTP under MVS.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5457E *Date Time #connection_number* **DBCSEtoA Conversion Error**

Explanation: A DBCS EBCDIC to ASCII conversion error occurred when transferring mail using the specified connection number. The body of the transferred mail contained invalid DBCS characters. DBCS characters in error are transmitted unconverted.

System Action: SMTP continues.

User or Operator Response: Use the connection number to determine the originator and destination of the invalid mail. The DEBUG statement may be configured in *hlq*.SMTP.CONFIG to aid in matching connection number to mail originator or destination. Check the mail body to see if it contains valid DBCS characters for the DBCS conversion type configured for the SMTP server. See *z/OS Communications Server: IP Configuration Reference*, and *z/OS Communications Server: IP User's Guide and Commands*, for more information about DBCS support in SMTP.

System Programmer Response: None.

Source Data Set: SMTPRPLY

Procedure Name: ResumeFillingDataBuffer

EZA5458E DBCS conversion specified, but no DBCS tables could be loaded

Explanation: DBCS conversion is configured for the SMTP server, but the required DBCS translation table could not be loaded. DBCS conversion is disabled.

System Action: The program continues.

User or Operator Response:

User or 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 more information about the loading and customizing of DBCS translation tables.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenOutputStreams

EZA5459E *date time #connection* **DBCSAtoE Conversion Error**

Explanation: A DBCS ASCII to EBCDIC conversion error occurred when transferring mail using the specified connection number. The body of the transferred mail contained invalid DBCS characters. DBCS characters in error are transmitted unconverted.

System Action: The program continues.

User or Operator Response: Use the connection number to determine the originator and destination of the invalid mail. The DEBUG statement may be configured in *hlq*.SMTP.CONFIG to aid in matching the connection number to the mail originator or destination. Check the mail body to see if it contains valid DBCS characters for the DBCS conversion type configured for the SMTP server. See *z/OS Communications Server: IP Configuration Reference*, and *z/OS Communications Server: IP User's Guide and Commands*, for more information about DBCS support in SMTP.

System Programmer Response: None.

Source Data Set: SMTPEVNT

Procedure Name: DoDbcsConversion

EZA5460I *date time* **BSMTP Helo Domain:** *sending host name host reply*

Explanation: If the debug option is on, this message specifies the domain name of the sending host, along with a reply indicating whether the HELO command is accepted.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPCMDS

Procedure Name: DoHelo

EZA5461I *date time* **TCP (***user domain name***) Helo Domain:** *host domain name user IP address*

Explanation: If you enter the HELO command while using a TCP connection to SMTP, this message specifies the domain name of your local host, as well as your domain name and IP address.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPCMDS

Procedure Name: DoHelo

EZA5462I *date time* **Restricting** *BSMTP/TCP MAIL FROM/RCPT TO* **: '***user name@domain.host name***'**

Explanation: The user is restricted from using SMTP. This message also indicates whether the connection is a batch connection or a TCP connection. For more information about restricted users, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPCMDS

Procedure Name: DoRestricted

EZA5463E *date time* **Unregistered BSMTP Spool File from** *userid@nodeid*

Explanation: An unregistered batch spool data set was received from the sender at the indicated node ID. This condition occurred because the sender is not registered in the DD data set security table. The BSMTP data set is not delivered.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: The sender should obtain authorization to use the secure gateway. To do so, the sender should follow the instructions contained in the SECURITY.MEMO data set, which is returned from the SMTP server just before this batch log entry.

Source Data Set: SMTPCMDS

Procedure Name: DoMail

EZA5464E *date time* **Unregistered BSMTP MAIL FROM:** *print path*

Explanation: SMTP sends this message to the batch job log to show that unregistered BSMTP mail was received from the address specified. The condition occurred because the sender is not registered in the DD data set security table. The BSMTP mail is not delivered.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: The sender should obtain authorization to use the secure gateway. To do so, the sender should follow the instructions contained in the SECURITY.MEMO data set, which is returned from the SMTP server just before this batch log entry.

Source Data Set: SMTPCMDS

Procedure Name: DoMail

EZA5465E *date time* **Deleting Spool File from undetermined origin**

Explanation: This message is sent to the SMTP batch log to show when a spool data set of an unknown origin is deleted. SMTP cannot accept an NJE spool data set if its origin cannot be determined. This message is generated because either the originator's user ID or the originator's node ID was missing. There are two ways to spool a data set to SMTP:

- The TSO XMIT command
- The batch SMTP commands with the IEBGENER utility in TSO/ISPF

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Determine why SMTP is receiving spooled data sets without the originator's user ID or the originator's node ID, and correct the error.

If you are using the TSO XMIT command, it must be set up correctly to pass a valid NJE user ID and a valid NJE node ID. For more information about the TSO XMIT command, see the *z/OS TSO/E Command Reference*.

If you are using the Batch SMTP interface, check your application that generates the SMTP commands for HELO and MAIL FROM. The MAIL FROM command must have the originator's user ID and the originator's host ID (equivalent to the originator's node ID.) See the *z/OS Communications Server: IP User's Guide and Commands* for more information about SMTP commands and batch SMTP.

Source Data Set: SMTPBTCH

Procedure Name: MakeBatchSMTPErrorFile

EZA5466E *date time* **Deleting Spool File from** *userid* **at** *nodeid*

Explanation: The spool data set from the specified user ID and node ID is being deleted because the user ID started with an asterisk (*).

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Determine the origin of the mail. Normally this message can be ignored.

Source Data Set: SMTPBTCH

Procedure Name: MakeBatchSMTPErrorFile

EZA5467E *date time* **Rejecting job number (***spool ID***) From** *userid* **at** *nodeid*

EZA5468E *date time* **Gateway is not enabled and the file is not from the local node (***node name***)**

Explanation: The job number was rejected for the indicated user ID and node ID either because the sender's address is in the RESTRICT list or because the TCP-to-NJE mail gateway is not enabled.

System Action: SMTP continues.

User or Operator Response: Check to ensure that the IEFSSNxx parmlib member has the correct JES node name specified and that the IKJTSO00 parmlib has the correct MVS node name specified on the TRANSREC/TRANSMIT statement.

System Programmer Response: Run the SMTPNJE program to configure the SMTP server to run as a mail gateway between TCP network users and users located on an NJE network attached to the local host. For more information about the RESTRICT list and the SMTPNJE program, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPBTCH

Procedure Name: InformUser

EZA5472I *date time* **Generating Note** *data set name* **as BSMTP REPLY**

Explanation: If the delivery type specified for the note command is "BATCH" and the LOG statement in the *hlq*.SMTP.CONFIG data set is enabled, SMTP sends a data set containing the note and any error message(s) to the sender when the connection specified cannot be established. This error message is also sent to the SMTP LOGFILE.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Perform™ a Resolver Trace for the SMTP server to determine why the connection could not be established. Take corrective actions as necessary, and resend the note. For more information about SMTP traces, see *z/OS Communications Server: IP Diagnosis*. If more help is required contact the IBM Software Support Center.

Source Data Set: SMTPQUEU

Procedure Name: MakeAddrBlok

EZA5473I *date time* **Generating Note** *note name* **as Error Notification**

Explanation: If the delivery type specified in the note command is not "BATCH" and the LOG statement in the *hlq*.SMTP.CONFIG data set is enabled, SMTP sends a note containing an error message to the sender when the connection specified cannot be established. This error message is also sent to the SMTP LOGFILE.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Perform a Resolver Trace for the SMTP server to determine why the connection could not be established. Take corrective actions as necessary, and resend the note. For more information about SMTP traces, see *z/OS Communications Server: IP Diagnosis*. If more help is required contact the IBM Software Support Center.

Source Data Set: SMTPQUEU

Procedure Name: MakeAddrBlok

EZA5474I *date time* **Received Note** *note name* **via BSMTP From** *printpath bytes* **bytes**

Explanation: If the sender specified the BATCH parameter in the note command and the LOG statement in the *hlq*.SMTP.CONFIG data set is enabled, this message is sent to the SMTP LOGFILE to indicate that the connection was established and that a data set containing the note of the size specified was received at the indicated date and time.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPQUEU

Procedure Name: MakeAddrBlok

EZA5475I *date time* **Received Note** *note name* **via TCP** *connection number* **From** *printpath bytes* **Bytes**

Explanation: If the sender did not specify the BATCH parameter in the note command and the LOG statement in the *hlq*.SMTP.CONFIG data set is enabled, this message is sent to the SMTP LOGFILE to indicate that the connection was established and that a note of the size specified was received at the indicated date and time.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPQUEU

Procedure Name: MakeAddrBlok

EZA5476I *date time* **Delivered Note** *note name* **to** *printpath*

Explanation: This message is sent to the SMTP LOGFILE to show that an SMTP note was delivered to the user ID at the node ID specified.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPQUEU

Procedure Name: MarkDeliveredRecipients

EZA5477E *date time* **Unable to Deliver Note** *note name* **to** *printpath*

Explanation: This message is sent to the SMTP LOGFILE to show that SMTP was unable to deliver the note to the user ID at the node ID specified.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Make sure that the recipient of the note is a user on an NJE or RSCS node connected to a TCPIP network at gateway_name.domain. Also check that the path address is correct. For more information, see *z/OS Communications Server: IP Diagnosis*.

Source Data Set: SMTPQUEU

Procedure Name: MarkDeliveredRecipients

EZA5478E *date time* **Unregistered RCPT TO:** *printpath*

Explanation: This message is sent to the SMTP LOGFILE to show that the indicated user is not a registered secure gateway user.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Update the DD data set security table as necessary. For more information, see *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: SMTPRES

Procedure Name: ProcessLocal, ProcessRQR

EZA5480I *date time connection* **TcpOpen returned connection still in use.**

Explanation: SMTP attempted to open the TCP connection that is displayed in the message. The TCP connection is opened and in use.

System Action: SMTP creates a timer for the connection and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: OpenConnection

EZA5481I *date time connection* **Received ConnectionStateChange Notification with NewState = Open**

Explanation: SMTP has received a TCP/IP event notification of CONNECTIONstateCHANGE. The SMTP TCP connection displayed in the message has been changed to OPEN.

System Action: SMTP changes the TCP connection state to open and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5482I *date time connection* **Connection already open, StateSet =** *connection state*

Explanation: SMTP has received a TCP/IP event notification of CONNECTIONstateCHANGE. SMTP is ignoring the request to open the connection because it is already open.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5483I *date time connection* **Closing old** *direction* **Connection**

Explanation: SMTP has received a TCP/IP event notification of CONNECTIONstateCHANGE. SMTP will close the old TCP connection.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5484I *date time connection* **Aborting new** *direction* **Connection**

Explanation: SMTP has received a TCP/IP event notification of CONNECTIONstateCHANGE. SMTP will close the TCP connection.

System Action: SMTP issues a TCPABORT against the connection and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5485I *date time connection* **Ignoring ConnectionStateChanged Notification for Closed Connection.**

EZA5486I *date time connection* **NewState =** *newstate*

Explanation: SMTP has received a TCP/IP event notification of CONNECTIONstateCHANGE for a closed connection. SMTP will ignore the request.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5487I *date time QueryNumber* **Popping request from Wait to SendQ**

Explanation: This debug message is displayed when SMTP is attempting to resolve a domain name. SMTP has waited the amount of time defined by the RESOLVERRETRYINT and will retry the request to the name server.

System Action: SMTP sends the request and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5488I *date time QueryNumber* **ReSetting Wait Timer** *seconds*

Explanation: This debug message is displayed when SMTP resets the wait timer to the value defined by RESOLVERRETRYINT.

System Action: SMTP resets the timer and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5489I *date time QueryNumber* **Popping request from Wait to Send**

Explanation: This debug message is displayed when SMTP is attempting to resolve a domain name. SMTP has waited the amount of time defined by the RESOLVERRETRYINT and will retry the request to the name server.

System Action: SMTP sends the request and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5490I *date time QueryNumber* **Resseting Retry Timer:** *seconds left* **seconds**

Explanation: This debug message is displayed when SMTP is attempting to resolve a domain name. SMTP resets the retry timer.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5491I *date time* **UDP packet arrived from:** *foreign_host data_legnth* **/** *full_legnth* **bytes.**

Explanation: This debug message is displayed when SMTP has received a UDP packet. The name of the host that sent the packet, data length and the full length of the packet are displayed.

System Action: The Packet is received and SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: MainLoop

EZA5492I *date time#errconnection* **Looping mail transfer to** *badspoolfile*

Explanation: SMPT has detected looping mail. The mail file is sent to the file BADSPOOLFILE.

System Action: SMTP sends the mail file to the BADSPOOLFILE and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPBTCH

Procedure Name: MakeBatchSMTPErrorFIle

EZA5496I *date time#connection* **Syntax Error: args after '***command* **command.** *command line*

Explanation: The SMPT command was entered with arguments. An error was detected because the command entered does not accept arguments.

System Action: SMTP ignores the command and continues.

User or Operator Response: Reenter the command without arguments. For more information about SMPT commands, see the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: SMTPCMDS

Procedure Name: DoCommand

EZA5497I *date time* **Enqueuing file** *data set name* **recipient** *recipient number* **on** *ip address*

Explanation: This debug message is displayed when a mail file is enqueued. The recipient number on the queue and the IP address are displayed. For more information on queued mail, refer to the *z/OS Communications Server: IP User's Guide and Commands*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5498I *date time#connection* **552 Too many hops - threshold exceeded**

Explanation: The SMTP file has hopped more than 25 times. The SMTP mail file will not be delivered.

System Action: SMTP returns the mail file to the sender and continues.

User or Operator Response: Verify that the recipient exists and resend.

System Programmer Response: The actual failing piece of mail is saved in a data set with the a naming convention of MailFileDSPPrefix.Conn*xxx*.LOOPMAIL, where the value of MailFileDSPPrefix can be found in the SMTP configuration data set and *xxx* represents the connection number which received the message. Use this data set to determine problems with the recipient address. This data set can then be deleted after the problem has been resolved.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5499I *date time#connection* **451 Local error: Disk Full**

Explanation: SMTP was unable to store the mail file because there was not enough disk space available.

System Action: SMTP returns the mail file to the sender and continues.

User or Operator Response: Increase the volume size for SMTP.

System Programmer Response: None.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5500I *date time#connection* **451 Local error: Allocation Error.**

Explanation: The SMTP attempted to allocate a file and received an allocation error.

System Action: SMTP returns a reply code to the sender and discards the SMTP file.

User or Operator Response: Verify that the volume is available.

System Programmer Response: None.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5501I *date time#connection* **552 Local error: Mail file too large.**

Explanation: The SMTP attempted to receive a mail file that is too large.

System Action: SMTP returns a reply code to the sender and discards the SMTP file.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPEVNT

Procedure Name: StoreData

EZA5502I *date time* **Enqueuing file** *dataset* **recipient** *recipient* **on** *address*

Explanation: SMTP is placing the indicated data set in the queue to the specified recipient.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPQUEU

Procedure Name: EnqueueTCPRecipient

EZA5503I *date time* **Processing Sender Address:** *path*

Explanation: SMTP is processing the path for the indicated sender address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: ProcessSender

EZA5504I *date time* **Sender converted to:** *address*

Explanation: SMTP has converted the sender address to the indicated IP mail address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: ProcessSender

EZA5505I *date time#connection* **Rcpt to unknown host:** *next_hop*

Explanation: SMTP has received mail for a host that it does not recognize. The mail will be sent to the indicated address for further resolution.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMPTRES

Procedure Name: ProcessRSCS

EZA5506I *date time* **Resolving Recipient Address:** *address*

Explanation: SMTP is resolving the address of the indicated mail recipient to an IP mail address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: ProcessRQR

EZA5507I *date time* **MX records left** *preference MXname*

Explanation: This message indicates the preference and mailbox name for the remaining mail records.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: ProcessRQR

EZA5508I *date time* **Not Found in Host Table:** *address*

Explanation: SMTP did not find the indicated mail address in its Host Tables. SMTP will attempt to resolve the address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUp

EZA5509I *date time* **Not found in Host Table:** *host_name*

Explanation: SMTP did not find the indicated host name in its Host Table. SMTP will attempt to resolve the host name.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUp

EZA5510I *date time* **Found in Host Table:** *address*

Explanation: SMTP found the indicated address in its Host Table.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUP

EZA5511I *date time* **Not Found in Host Table:** *name*

Explanation: SMTP could not find the indicated name in the Host Table. SMTP attempts to resolve the name.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUP

EZA5512I *date time* **Found in Host Table:** *name*

Explanation: SMTP found the indicated name in the Host Table.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUP

EZA5513I *date time* **Canonicalizing To:** *canonical_name*

Explanation: SMTP is converting an address to the indicated canonical name.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUp

EZA5514I *date time* **Found in Host Table:** *host name*

Explanation: SMTP found the indicated host name in the host table.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUp

EZA5515I *date time* **Canonicalizing To:** *canonical_name*

Explanation: SMTP is converting an address to the indicated canonical name.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUp

EZA5516I *date time* **Unknown Host Name:** *name*

Explanation: SMTP has received mail for the indicated host, but does not have the host name in the host table.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the SMTP host table if necessary.

Source Data Set: SMTPRES

Procedure Name: HostTableLookUP

EZA5517I *date time*#*query_number* **Unable to resolve request for** *number* **days**

Explanation: SMTP has been unable to resolve a query for an address for the indicated number of days. SMTP returns the query to the sender.

System Action: Processing continues.

User or Operator Response: Make sure that the host requested exists on the network.

System Programmer Response: Assist the user as necessary.

Source Data Set: SMTPRES

Procedure Name: SendRQR

EZA5518I *date time*#*query_number* **Adding Request to Retry Queue**

Explanation: SMTP was unable to resolve a query, and is placing the query on the queue to be retried.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: SentRQR

EZA5519I *date time*#*query_number* **Setting Retry Timer:** *number* **seconds**

Explanation: SMTP is setting the retry timer, which governs the frequency with which requests are retried, to the indicated value.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: SendRQR

EZA5520I *date time*#*query_number* **UDP Query Sent, Try:** *number* **to NS** *name_server* **:=** *address*

Explanation: SMTP is sending a query to the indicated name server for resolution. The number of times SMTP has tried to resolve the query is indicated in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: SendRQR

EZA5521I *date time*#*query_number* **Adding Request to Wait Queue**

Explanation: SMTP was unable to resolve the indicated query, and is placing it on the WAIT queue to be retried.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: SendRQR

EZA5522I *date time*#*query_number* **Setting Wait Timer:.** *value*

Explanation: SMTP is setting the WAIT timer, which specifies the length of time SMTP waits before retrying a query, to the indicated value.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: SentRQR

EZA5523I *date time* **Invalid Namerserver Header Record**

Explanation: SMTP has received a name server record with a header that contains no data. The record is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5524I *date time#query_number* **Remote Nameserver down/unable to perform recursion**

Explanation: A remote name server to which SMTP sent a query for resolution is either not running or unable to perform recursion. The query is queued and retried.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the remote name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5525I *date time#query_number* **Discarding Name Server Response with Format Error**

Explanation: SMTP received an incorrectly formatted response from a name server. The response is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5526I *date time#query_number* **Truncation Bit set in Nameserver Response**

Explanation: SMTP received a response from a name server with a truncation bit indicating that the response has been truncated.

System Action: The response is discarded and the query is repeated.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5527I *date time#query_number* **Discarding Invalid Nameserver Response**

Explanation: SMTP has received a response to a name server query that either does not contain a valid name or has a data length error on the response. The response is discarded and the query is repeated.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5528I *date time#query_number* **Unable to match Nameserver Response with Query**

Explanation: SMTP was unable to match a response from a name server with a query. The response is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5529I *date time#query_number* **Discarding Invalid Nameserver Record**

Explanation: SMTP received a response to a name server query that did not contain a valid name. The response is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5530I *date time#query_number* **Discarding Invalid CNAME Record**

Explanation: SMTP received a response to a mailbox type query that did not contain a valid name. The response is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5531I *date time#query_number* **Throwing away circular CNAME record:** *name*

Explanation: SMTP received a circular response to a CNAME type query. A circular response points back to itself, giving the canonical name instead of the address. The response is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5532I *date time#query_number* **Canonicalized with:** *canonical_name*

Explanation: SMTP has converted an address to the indicated canonical name.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5533I *date time#query_number* **Discarding Invalid MX Record**

Explanation: SMPT received a response to a mailbox query that did not contain a valid mailbox name. The response is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact the administrator for the name server and inform them of the problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5536I *date time*#*query_number* **Ignoring Non-Authoritative response**

Explanation: SMTP received a non-authoritative response to a name server query. Non-authoritative responses point to a name server that has authority for the correct zone. SMTP cannot send mail to the name server, so the non-authoritative answer is ignored. SMTP will send a name server query to determine the proper name server to use.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5537I *date time* **Converting** *dataset* **ADDRBLOK to new format**

Explanation: SMTP is converting the address block for the indicated data set to a new format.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRFMT

Procedure Name: ReformatAddrBlok

EZA5539I *date time* **Restricted SMSG command:** *command* **from:** *userID* **at:** *node*

Explanation: The indicated user issued an authorized SMSG command. The command will be executed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPSMSG

Procedure Name: ProcessSMSG

EZA5542I *date time*#*query_number* **Ignoring Referral response**

Explanation: SMTP received a referral response to a name server query. SMTP does not do recursive processing and relies on the name server for recursive processing. The referral is ignored.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Inform the name server's administrator of the recursion problem.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5543I *date time*#*query_number* **Unable to resolve MX host name** *hostname*

Explanation: SMTP received an MX record for this host but did not receive the A record information. SMTP reissues the request to obtain the A information for the specified host name from the name server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: NameServerToRQR

EZA5544E Unable to RENAME file *data set name* **to** *data set name* **rc=***rc*

Explanation: An attempt by SMTP to rename a data set failed. This may cause SMTP to have problems with processing the mail.

System Action: Processing continues.

User or Operator Response: Refer this message to your system programmer.

System Programmer Response: Check the system log for message IEC614I. Use the diagnostic information provided and refer to *MVS/DFP V3R3 Diagnostic Reference*. If message IEC614I is not present, check for a return code of 408. This indicates that the new data set name already exists but is not cataloged. Locate and delete the data set.

Source Data Set: SMMVSUB

Procedure Name: svc202r

EZA5545E *date time addrblokname***.ADDRBLOK does not exist. No Mail Error process is attempted. RC =** *rc*

Explanation: SMTP tried to build an error file, but the original SMTP note, represented by the data set *addrblokname*.ADDRBLOK, has not been allocated.

date is the date in the form **mm/dd/yy**

time is the time in the form **hh:mm:ss**

addrblokname is the name of the SMTP data set

rc is the return code from the program that detected the error

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Check your system console to ensure that another application or defragmentation utility did not access or delete the SMTP data set while the SMTP server was active. Data set contention can cause this problem. To avoid this problem, configure the application to bypass data sets with the SMTP high level qualifier.

Source Data Set: SMTPUTIL

Procedure Name: MakeErrorFile

EZA5546E *date time* **Zero bytes were read from** *addrblokname***.ADDRBLOK. No Mail Error process is attempted. RC =** *rc*

Explanation: SMTP tried to build an error file, but the original SMTP note, represented by the data set *addrblokname*.ADDRBLOK, is empty. Therefore, the sender's address cannot be retrieved.

date is the date in the form **mm/dd/yy**.

time is the time in the form **hh:mm:ss**.

addrblokname is the name of the SMTP data set.

rc is the return code from the program that detected the error.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Check your system console to ensure that another application or defragmentation utility did not access or delete the SMTP data set while the SMTP server was active. Data set contention can cause this problem. To avoid this problem, configure the application to bypass data sets with the SMTP high level qualifier.

Note: Before you start the SMTP server again, delete *addrblokname*.ADDRBLOK.

Source Data Set: SMTPUTIL

Procedure Name: MakeErrorFile

EZA5547I *date time* **Processing Path String:** *pathstring* **and length =** *length*

Explanation: This message displays the path string built by SMTP. This message will be displayed only if resolver trace is active.

date is the date in the form **mm/dd/yy**.

time is the time in the form **hh:mm:ss**.

pathstring is the path string built by SMTP.

length is the path string length. The maximum length allowed is 256 characters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRES

Procedure Name: PrintPath

EZA5548E Unable to open existing file *FullDataSetName***, rc=** *ReturnCode***. SMTP Terminating.**

Explanation: SMTP attempted to open the data set specified in the error message which already existed and received an error from the operating system.

FullDataSetName is the name of the data set.

ReturnCode is the value of the return code.

System Action: SMTP ends.

User or Operator Response: None

System Programmer Response: If return code is 28, check to see if there is data set contention for the data set name displayed in the error message. SMTP must have access to all its data sets to perform its function. Therefore, if a defragmentation utility (or any other application) is actively accessing SMTP data sets while the SMTP server is active, a data-set contention problem could occur. To avoid this problem, configure the application to bypass data sets with the SMTP high level qualifier. The high level qualifier is set by the SMTP configuration parameter MAILFILEDSPREFIX.

Source Data Set: SMMVSUB

Procedure Name: svc202

EZA5549I A USER EXIT HAS BEEN ACTIVATED FOR THIS INVOCATION OF SMTP

Explanation: A user supplied exit has been located, loaded, and activated for this execution of the SMTP program.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZABB01X

Procedure Name: None.

EZA5550I *date time* **EXIT REJECTS - CONNECTION** *connection_number* **ACTIONCODE** *action*

Explanation: The user exit has chosen to reject an inbound mail item. This message is written to the logfile, if the logfile option is active.

date and *time* indicate when the connection was rejected.

connection_number is the unique identifier of this connection.

action is the action code that was used when the exit was called. The action codes are defined in the EZBZSMTP macro.

System Action: The partner SMTP application is notified that service is denied due to a user supplied exit and processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZABB020

Procedure Name: DoExit

EZA5551I *connection_ number*< **EXITPARMS VER** *version* **ACTN** *actioncode* **USER TOKEN** *value* **IPADDR** *ipaddr* **DLEN** *length*

Explanation: This parmlist was passed to the user supplied SMTP exit for the connection number specified. This message will be written to the debug file, if the debug option is active. The less than symbol (<) indicates data flowing to the user supplied exit.

connection_number is the unique identifier of this connection.

version is the version number of the parmlist.

actioncode is the action code for which the user supplied exit was called. The action codes are defined in the EZBZSMTP macro.

value is a 4-byte value supplied by the user.

ipaddr is the IP address of the partner SMTP.

length is the length of the data in the buffer.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZABB020

Procedure Name: DoExit

EZA5552I *connection_number*> **USER EXIT RETCODE** *returncode*

Explanation: The displayed return code was received from the user supplied SMTP exit for the connection number specified. This information is written to the debug file, if the debug option is active.

connection_number is the unique identifier of this connection.

returncode is a user-supplied value. The return code should be selected from those defined in the EZBZSMTP macro.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZABB020

Procedure Name: DoExit

EZA5553I *date time* **SitePtr =** *siteptr* **File_Queue** *dataset_name* **NextQueued =** *next_siteptr*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

siteptr is the pointer to the current site for the mail.

dataset_name is the file name pointed to by File_Queue. This data set contains a single piece of mail.

next_siteptr is the pointer to the next site for the mail.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: You can use the *dataset_name* and other messages generated in the trace to determine if the data set has been placed on the retry queue or is still on the active SMTP queue working with secondary IP address. Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5554I *date time* **NextSite =** *nextsite* **File_Queue** *dataset_name* **NextQueued =** *next_nextsite*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

nextsite is the pointer to the next site for the mail.

dataset_name is the file name pointed to by File_Queue. This data set contains a single piece of mail.

next_nextsite is the pointer to the next site for the mail.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: You can use the *dataset_name* and other messages generated in the trace to determine if the data set has been placed on the retry queue or is still on the active SMTP queue working with secondary IP address. Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5555I *date time* **Retry_Queue =** *retryqueue_head* **File_Queue** *dataset_name* **NextQueued =** *retryqueue_nextsite* **where =** *timing_sequence*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

retryqueue_head is the pointer to the head of the retry queue for mail.

dataset_name is the file name pointed to by File_Queue. This data set contains a single piece of mail.

retryqueue_nextsite is the pointer to the next site for the mail on the retry queue.

If *timing_sequence* is **1**, this is the retry queue values prior to adding or deleting from the queue. If *timing_sequence* is

2, this is the retry queue value after changes have been made.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: You can use the *dataset_name* and other messages generated in the trace to determine if the data set has been placed on the retry queue. *timing_sequence* applies to both this message and EZA5556I. Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5556I *date time* **Retry_QTail =** *retryqueue_tail* **File_Queue** *dataset_name* **NextQueued =** *retryqueue_tail_nextsite*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

retryqueue_tail is the pointer to the tail of the retry queue for mail.

dataset_name is the file name pointed to by File_Queue. This data set contains a single piece of mail.

retryqueue_tail_nextsite is the pointer to the next site for the mail on the retry queue tail. This should always be zero.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: You can use the *dataset_name* and other messages generated in the trace to determine if the data set has been placed on the retry queue. This message immediately follows message EZA5555I. Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5557I *date time* **SMTP_Queue =** *SMTPqueue_head* **File_Queue** *dataset_name* **NextQueued =** *SMTPqueue_nextsite* **where =** *timing_sequence*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

SMTPqueue_head is the pointer to the head of the SMTP queue for mail.

dataset_name is the file name pointed to by File_Queue. This data set contains a single piece of mail.

SMTPqueue_nextsite is the pointer to the next site for the mail on the SMTP queue.

If *timing_sequence* is **1**,, this is the SMTP queue values prior to adding or deleting from the queue. If *timing_sequence* is **2**, this is the SMTP queue after changes have been made.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: You can use the *dataset_name* and other messages generated in the trace to determine if the data set has been placed on the active SMTP queue working with secondary IP address. *timing_sequence* applies to both this message and EZA5558I. Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5558I *date time* **SMTP_QTail =** *SMTPqueue_tail* **File_Queue** *dataset_name* **NextQueued =** *SMTPqueue_tail_nextsite*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

SMTPqueue_tail is the pointer to the tail of the SMTP queue for mail.

dataset_name is the file name pointed to by File_Queue. This data set contains a single piece of mail.

SMTPqueue_tail_nextsite is the pointer to the next site for the mail on the SMTP queue tail. This should always be zero.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: You can use the *dataset_name* and other messages generated in the trace to determine if the data set has been placed on the active SMTP queue working with secondary IP address. This message immediately follows message EZA5557I. Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5559I *date time* **SitePtr =** *siteptr* **to be disposed NextQueued =** *nextsite*

Explanation: This diagnostic message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever the TCPDATA file has RESOLVER traces active.

date is the current date.

time is the current time.

siteptr is the pointer currently being disposed of.

nextptr is the pointer to the next site for the mail. Should always be zero.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: *siteptr* shows what pointer is being disposed of (freed). Review the messages and their sequence to determine if you have a problem. If there is a problem, collect these traces and contact the IBM Software Support Center.

Source Data Set: SMTPEVNT

Procedure Name: CloseCompletedEvent

EZA5560I *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 multple times to include the completer text from Dynamic Allocation.

error_message is text of the message produced by DYNALLOC .

System Action: SMTP might halt or continue depending on the preceding SMTP error message.

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

System Programmer Response: Use the text of this message and the documentation describing Dynamic Allocation (DYNALLOC) in the *z/OS MVS Programming: Authorized Assembler Services Guide* to determine the cause of the error and respond as indicated. Most of these problems are DASD management related, so stop the SMTP started task, correct the problem, then restart the task.

Source Data Set: SMTP_s99error

Procedure Name: SMMVSUB

EZA5561I RESOLVERUSAGE NO specified without a forwarding mailer

Explanation: SMTP resolver usage was disabled by the RESOLVERUSAGE NO statement in the *hlq*.SMTP.CONFIG data set and a remote mailer to handle unknown mail has not been specified. All non-local mail received by SMTP is returned as undeliverable.

System Action: SMTP does not attempt to resolve non-local host names. Any non-local mail received by SMTP is returned as undeliverable.

User or Operator Response: Notify system the programmer.

System Programmer Response: Specify a remote mailer where unknown mail will be forwarded by specifying the IPMAILERADDRESS statement or the MAILER...UNKNOWN statement. For information about the IPMAILADDRESS and the MAILER...UNKNOWN statements, see the *z/OS Communications Server: IP Configuration Guide*.

Source Data Set: SMTP

Procedure Name: DoFinalChecks

EZA5562I Resolver Usage *parm* Specified

Explanation: RESOLVERUSAGE was specified in the *hlq*.SMTP.CONFIG data set. The new value *parm* will change the current value for this configurable setting.

parm is either YES to attempt resolving non-local host names or NO to inhibit the resolving of non-local host names.

System Action: If the final setting is NO, SMTP will not attempt to resolve any non-local host names. Mail will be forwarded if the IPMAILERADDRESS statement or the MAILER...UNKNOWN statement has been specified in the *hlq*.SMTP.CONFIG data set. For information about the IPMAILADDRESS and the MAILER...UNKNOWN statements, see the *z/OS Communications Server: IP Configuration Guide*.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTP

Procedure Name: ProcessConfigFile

EZA5563I Incorrect or missing operand associated with statement RESOLVERUSAGE *incorrect_operand*

Explanation: The only valid operands for the RESOLVERUSAGE statement are YES and NO. Either YES or NO must be specified.

incorrect_operand is the incorrect operand that was found. If no operand was specified, this will be blank.

System Action: SMTP ends.

User or Operator Response: Notify the system programmer.

System Programmer Response: Specify the correct operand for the RESOLVERUSAGE statement and restart the SMTP job to pick up the new settings in the SMTP configuration data set.

Source Data Set: SMTP

Procedure Name: Error

EZA5564I *date time* **Processing spool file** *filename*

Explanation: This message is written to the SMTP sysdebug data set specified in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever debug is active and SMTP is processing data from the JES spool.

date is the current date.

time is the current time.

filename is the spool file name being processed by SMTP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ServiceReaderMVS

Procedure Name: SMTPBTCH

EZA5565I *date time* **Read spool file** *filename* **sent by** *userid* **at** *nodeid* **using TSO XMIT**

Explanation: This message is written to the SMTP sysdebug data set specified in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever debug is active and SMTP is processing data from the JES spool. The message indicates that the spool file was sent using the TSO/E TRANSMIT(XMIT) command. This command puts certain information in its headers indicating who and where the XMIT command was issued from. The spool file is transmitted with headers, built by XMIT, that identify the originating node and user ID.

date is the current date.

time is the current time.

filename is the name of the spool file being processed by SMTP.

userid is the user identification taken from the XMIT headers.

nodeid is the NJE node identification taken from the XMIT headers.

System Action: SMTP continues

User or Operator Response: If *userid* or *nodeid* is incorrect then contact the system programmer.

System Programmer Response: If *userid* or *nodeid* is incorrect then check the XMIT command setup parameters.

Source Data Set: ServiceReaderMVS

Procedure Name: SMTPBTCH

EZA5566I *date time* **Read spool file** *filename* **sent by a batch job**

Explanation: This message is written to the SMTP sysdebug data set specified in the SYSDEBUG statement in *hlq*.SEZAINST(SMTPPROC). This message is generated whenever debug is active and SMTP is processing from the JES spool. The message indicates that the spool file was sent by a batch job.

date is the current date.

time is the current time.

filename is the name of the spool file being processed by SMTP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ServiceReaderMVS

Procedure Name: SMTPBTCH

EZA5567I *filename*.**ADDRBLOK was incomplete and was deleted.**

Explanation: This message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST (SMTPPROC). This message is generated when SMTP detects a partial control block data set (*filename*.ADDRBLOK) exists. The control block data set must contain the following records:

- record 1 - header information.
- record 2 - sender information.
- record 3 - recipient information. At least one recipient record must be present.

If SMTP detects a partial data set it will delete *filename*.ADDRBLOK and any corresponding *filename*.NOTE data set. Partial data sets might be created by failures to get space on the DASD.

filename.ADDRBLOK is the name of the data set being deleted by SMTP.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Ensure that there is enough DASD space available for SMTP to continue.

Source Data Set: EnqueueForDelivery

Procedure Name: SMTPQUEU

EZA5568E UNABLE TO ALLOCATE ADDRBLOK DATA SET™ CORRECTLY.

Explanation: This message is written to the SMTP output data set specified in the OUTPUT statement in *hlq*.SEZAINST (SMTPPROC). SMTP cannot allocate enough space for its control block data set (*.ADDRBLOK) . This problem might be caused by not enough available DASD space.

System Action: SMTP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: If SMTP did not end, then use the MVS stop command to end SMTP. Ensure that the DASD used by SMTP is defined properly and has enough space available. Correct any DASD problem and restart SMTP.

Source Data Set: MakeAddrBlok

Procedure Name: SMTPQUEU

EZA5569I No Source Routing : *parameter*

Explanation: The NOSOURCEROUTE statement in the *hlq*.SMTP.CONFIG file has been set. Depending on the *parameter* following the statement, SMTP will modify its behavior regarding source routing addressing strings that can be passed on both the MAIL FROM: and RCPT TO: SMTP commands.

parameter is an input string whose valid options are DISABLED, MAILCMD, RCPTCMD or ENABLED.

See the *z/OS Communications Server: IP Configuration Reference* for more information about the NOSOURCEROUTE configuration statement.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: OpenOutputStreams

Procedure Name: SMTP

EZA5570E INCORRECT VALUE *parameter* SPECIFIED FOR NOSOURCEROUTE STATEMENT - THE DEFAULT VALUE OF DISABLED WILL BE USED

Explanation: The input parameter following the NOSOURCEROUTE statement is invalid. The default value of DISABLED will be used instead. This means that the SMTP server will generate and pass source route addressing strings on both the MAIL FROM: and the RCPT TO: SMTP commands.

parameter shows a maximum of ten characters of the invalid input parameter.

See the *z/OS Communications Server: IP Configuration Reference* for more information about the NOSOURCEROUTE configuration statement.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: Change the parameter associated with the NOSOURCEROUTE configuration statement to a valid parameter. Stop and restart SMTP to activate the new parameter setting.

Source Data Set: ProcessConfigFile

Procedure Name: SMTP

EZA5571E *date time* SPOOL FILE *jobid* WAS DELETED DUE TO RECORD COUNT OF ZERO

Explanation: The JES spool job identified in this message was intentionally deleted by the SMTP code because it contained a total record count (SSSOLNCT) of zero. See the *z/OS MVS Using the Subsystem Interface* for more information about the JES interface used by SMTP.

date is the current date stamp.

time is the current time stamp.

jobid is the JES spool identifier (SSSOJOBI).

System Action: SMTP deletes the JES spool file. Processing continues.

User or Operator Response: None.

System Programmer Response: Check the applications that use the JES spool for SMTP. The most probable cause for these files being created with a zero record count is an application abend.

Source Data Set: ServiceReaderMVS

Procedure Name: SMTPBTCH

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: SMTP continues.

User or Operator Response: Specify a destination and resend the note.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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 EXEC halts. SMTP continues.

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

System Programmer Response: Increase the region size and restart SMTPNOTE.

Source Data Set: SMTPNOTE

Procedure Name: main.

EZA5577I NOTE CANCELLED

Explanation: A note created using SMTPNOTE has been canceled by the user.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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: SMTP continues.

User or Operator Response: Notify the system programmer of this message.

System Programmer Response: Increase the region size and restart SMTPNOTE.

Source Data Set: 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. This message will be accompanied by messages from the operating system giving further information about the error.

System Action: The data is not transmitted. SMTP continues.

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

System Programmer Response: Use the messages from the operating system to determine the cause of the error, and respond as indicated.

Source Data Set: 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: SMTP continues.

User or 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.

Source Data Set: 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: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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: SMTP continues.

User or 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 similiar 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*.

Source Data Set: 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: SMTP continues.

User or 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 for Catalogs*.

Source Data Set: 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: SMTP continues.

User or 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: None.

Source Data Set: 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: SMTP continues.

User or 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: None.

Source Data Set: 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: SMTP continues.

User or Operator Response: Submit only one of these parameters when using SMTPNOTE.

System Programmer Response: None.

Source Data Set: 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: SMTP continues.

User or 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: None.

Source Data Set: 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: SMTP continues.

User or 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: None.

Source Data Set: 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: SMTP continues.

User or 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: None.

Source Data Set: SMTPNOTE

Procedure Name: main.

EZA5590S UNKNOWN ERROR CODE '*errcode*'

Explanation: SMTPNOTE has encountered an error that it does not recognize.

System Action: SMTP continues.

User or 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.

Source Data Set: SMTPNOTE

Procedure Name: main

EZA5600I Correct form is:

EZA5602I SmtpNJE MVSDSName <(SpoolSystemType>

EZA5603I where MVSDSName is the dataset name (including membername, if

EZA5604I any) of the JES2 or JES3 initialization stream, which is used as

EZA5605I input to SMTPNJE. This dataset must be cataloged. If the

EZA5606I dataset is not enclosed in quotes, the userid will prepended

EZA5607I to it. SmtpNJE will create the file: userid.SMTPNJE.HOSTINFO as

EZA5608I output, which should be renamed to the dataset name pointed to by

EZA5609I the SMTPNJE DD of the SMTP server.

Explanation: This is the combined help message displayed when a "?" is entered on the command line with SMTPNJE. This message is also displayed when an incorrect data set name is specified, or if there is a problem opening a data set. Finally, it is also displayed when incorrect form is used when specifying an optional parameter while initiating SMTPNJE. For more information about SMTPNJE, see the *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: GiveHelp

EZA5611E Unrecognized parameters: '*parameter***'**

Explanation: The data set name and any additional parameters should be seperated by an open parenthesis: "(".

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: Rerun the SMTPNJE program using the correct, specified format. See the *z/OS Communications Server: IP Configuration Reference* for information.

Source Data Set: SMTPRSCS

Procedure Name: ProcessCommandLine

EZA5612E Unable to find input file: *data_set*

Explanation: The data set name specified as the program invocation argument cannot be found on any MVS volume.

System Action: SMTPNJE halts.

User or Operator Response: None.

System Programmer Response: Check that SMTPNJE has access to the data set on one of the MVS volumes.

Source Data Set: SMTPRSCS

Procedure Name: ProcessCommandLine

EZA5613W Warning: Duplicate entry: *node*

Explanation: A duplicate name exists in the input data set being processed. The duplicate is ignored.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: If the input data set is in error, remove the duplicate name entry.

Source Data Set: SMTPRSCS

Procedure Name: Insert

EZA5614E Invalid system type specified: *type*

Explanation: An incorrect optional SpoolSystemType was entered at the command line. The only valid options are RSCS, JES2, or JES3. Use the form: SMTPNJE *data_set_name* (*JES2 or JES3*. The default is JES2.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: Reenter the command with a valid SpoolSystemType and the correct form.

Source Data Set: SMTPRSCS

Procedure Name: ParseOptionalPart

EZA5615E Unrecognized options: *option*

Explanation: SMTPNJE found additional characters in the option string after a valid option was found.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: Rerun the SMTPNJE program using the correct specified format. See the *z/OS Communications Server: IP Configuration Reference* for information.

Source Data Set: SMTPRSCS

Procedure Name: ParseOptionalPart

EZA5616E Invalid DSName Specifications: *data set*

Explanation: A data set name was entered that exceeds 44 characters in length. The incorrect data set name is indicated at the end of the message.

System Action: SMTPNJE halts.

User or Operator Response: None.

System Programmer Response: Correct the data set name, and rerun the program.

Source Data Set: SMTPRSCS

Procedure Name: ProcessCommandLine

EZA5617I Entries in the NJE Hash Table: *number of entries*

Explanation: Displays the number of nodes listed in the hash table.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: AmpxFree

EZA5618I Length of Hash Table: *length*

Explanation: Displays the length of the hash table.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: AmpxFree

EZA5619I Average seeks per Entry: *result*

Explanation: Displays the number of searches for a name in the hash table per number of entries.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: AmpxFree

EZA5620I Hash Table is: *percent*% **Full**

Explanation: Shows how full the hash table is in percentage.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: AmpxFree

EZA5622E Unable to find input file: *data set*

Explanation: The data set specified as the program invocation argument cannot be found on any MVS volume.

System Action: SMTPNJE halts.

User or Operator Response: Verify that the data set exists and that the user has access to it. volumes.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: GiveHelp

EZA5624E Maximum number of NJE nodes found. SMTPNJE only supports up

---

EZA5625E to *maxnodes***.**

Explanation: SMTPNJE found more that the specified number of JES2 or JES3 nodes allowed by the MaxNodes constant.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: Increase the MaxNodes constant in the SMTPRSCS Pascal program to the desired value, recompile the source, relink-edit the executable module, and run SMTPNJE again.

Source Data Set: SMTPRSCS

Procedure Name: EndofGame

EZA5627W JES Destination: *oldnode* **to :** *newnode*

Explanation: The specified JES node name is greater than the maximum of 8 characters and was truncated to the indicated name.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: Correct the JES node name to be 8 characters or fewer. Refer to the *z/OS Communications Server: IP Programmer's Reference* for more information about the JES node name.

Source Data Set: SMTPNJE

Procedure Name: ParseJES2initFile

EZA5628W JES Destination: *oldnode* **truncated to:** *newnode*

Explanation: The JES3 node name after NAME= is greater than the maximum of 8 characters and was truncated to the indicated name.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: Correct the JES3 node name to be 8 characters or fewer.

Source Data Set: SMTPRSCS

Procedure Name: ParseJES3InitFile

EZA5629E Error opening input file: *data set name*

Explanation: There was an error opening the specified data set on the command line when using the SMTPNJE *data set name* command.

System Action: SMTPNJE halts.

User or Operator Response: Verify the name of the data set specified and that it is accessible.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: ProcessCommandLine

EZA5630E Error opening output file: *set name*

Explanation: There was an error opening the file *userid*.SMTPNJE.HOSTINFO, which will be the data set containing the hash table. The attempt to allocate the file was unsuccessful.

System Action: SMTPNJE halts.

User or Operator Response: Verify the type of access you have to this data set. If necessary, ask the data set's owner for access.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: AmpxFree

EZA5631E No nodes found in *set name***. SMTPNJE data set will not be created.**

Explanation: There are no node names specifying which user's mail is supported, so SMTPNJE cannot create a hash table.

System Action: SMTPNJE continues.

User or Operator Response: Edit the data set that contains the names of the nodes. Verify that the data set contains NJE node definitions. For more information see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: SMTPRNJE

Procedure Name: Insert

EZA5632I SMTPNJE *version*

Explanation: Indicates the version of SMTPNJE in use.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: AmpxFree

EZA5633I Generic node ID not supported, ignoring *group_node RSCS_token*

Explanation: The system will not accept any node with an asterisk (*).

System Action: SMTPNJE continues.

User or Operator Response: Reenter a node that does not contain an asterisk.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: ProcessNode

EZA5641E IMBED not supported on MVS

Explanation: An IMBED statement was encountered while processing an RSCS style configuration data set. The IMBED is ignored.

System Action: SMTPNJE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SMTPRSCS

Procedure Name: IMBEDstatement

EZA5651N SIMPLIFIED CHINESE

Explanation: The SMTP server is configured to use Simplified Chinese if the DBCS statement in the *hlq*.SMTP.CONFIG data set is enabled. See also message EZA5263I. For more information on configuring the translation tables, see *z/OS Communications Server: IP Configuration Reference*.

System Action: SMTP continues.

User or Operator Response: None.

System Programmer Response: NONE.

Source Data Set: SMTP

Procedure Name: OPENOUTPUTSTREAMS

EZA5775I Usage: SNALINK tcpipname applid

Explanation: SNALINK was invoked with incorrect command line parameters. Correct command line syntax is shown.

System Action: SNALINK exits.

User or Operator Response: Retry the command with the correct parameters.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Retry the command with the correct parameters.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check that the SNALINK address space is defined correctly.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: SNALINK

Procedure Name: Main

EZA5779E Error *error* IN VTAM® OPEN

Explanation: SNALINK encountered an error issuing a VTAM* OPEN request.

System Action: SNALINK abends.

User or Operator Response: Check that VTAM is operational.

System Programmer Response: Check the SNALINK parameters for the correct VTAM_APPL_ID. Contact your VTAM administrator.

Source Data Set: SNALINK

Procedure Name: Main

EZA5780I Init complete, APPLID *id*, TCPIP id

Explanation: SNALINK is running.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: Main

EZA5782S TPEND exit called with reason *value*

Explanation: SNALINK exits with the indicated reason code.

System Action: SNALINK exits.

User or Operator Response: None.

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

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check the MVS system for system problems.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a probable TCPIP code problem. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This message indicates a probable TCPIP code problem. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check TCPIP parameters (MVS) for the correct TCPIP data set.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Increase the maximum connections allowed.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible TCPIP code problem. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: Tell your VTAM administrator.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Use the NETSTAT command to check that TCPIP started SNALINK.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell your VTAM administrator.

System Programmer Response: This error indicates a possible VTAM problem.

Source Data Set: SNALINK

Procedure Name: scipbind

EZA5799I Received BIND request for SNA receive session

Explanation: A BIND request has been received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell your VTAM administrator.

System Programmer Response: This error indicates a possible TCPIP code problem. Contact the IBM Software Support Center.

Source Data Set: SNALINK

Procedure Name: scipfoun

EZA5801I Got UNBIND for receive session

Explanation: An UNBIND request has been received.

System Action: The session is brought down.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: scipunbd

EZA5802I Got UNBIND for send session

Explanation: An UNBIND request has been received.

System Action: The session is brought down.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible TCPIP or VTAM code problem. Tell your VTAM administrator.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem.

Source Data Set: SNALINK

Procedure Name: vtamopnd

EZA5807I SNA send session established

Explanation: An SNA send session was established.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem. Tell your VTAM administrator.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem.

Source Data Set: SNALINK

Procedure Name: vtamopns

EZA5810I SNA receive session established

Explanation: An SNA receive session was established.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP error.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Tell your VTAM administrator.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible TCPIP code problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Restart SNALINK.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Source Data Set: 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.

User or Operator Response: Tell your VTAM administrator.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check your hardware or VTAM for errors.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check your hardware or VTAM for errors.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP problem. Tell your VTAM administrator.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Source Data Set: 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.

User or Operator Response: Check your hardware or VTAM for errors.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check your hardware or VTAM for errors.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP problem. Tell your VTAM administrator.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible VTAM or TCPIP code problem. See the VTAM console for more information.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell your VTAM administrator.

System Programmer Response: This error indicates a possible TCPIP code problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible TCPIP code problem.

Source Data Set: SNALINK

Procedure Name: dovtopnd

EZA5829I Accepting DLC path

Explanation: A BIND was received. SNALINK accepts a DLC connection to TCPIP.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible TCPIP code problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This error indicates a possible system problem.

Source Data Set: 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.

User or Operator Response: Check that TCPIP is operational.

System Programmer Response: This error indicates a possible TCPIP code problem.

Source Data Set: SNALINK

Procedure Name: doiucrcv

EZA5833E Error *error* **severing DLC path**

Explanation: SNALINK encountered an error severing a DLC path.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the *error* portion of this message.

Source Data Set: SNALINK

Procedure Name: dosever

EZA5834E Error *error* **in VTAM CLOSE**

Explanation: SNALINK encountered an error closing an ACB.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the *error* portion of this message.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check the error indication.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check the error indication.

Source Data Set: 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 retries, ends the session, or ignores the error and continues processing.

User or Operator Response: None.

System Programmer Response: Tell your VTAM administrator.

Source Data Set: 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 retries, ends the session, or ignores the error and continues processing.

User or Operator Response: None.

System Programmer Response: Tell your VTAM administrator.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: prtsense

EZA5840E SSENSEO, SSENSMO, USENSEXO: *value*

Explanation: This message displays Sense information.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: prtsense

EZA5841E ANALYZE returns *value*

Explanation: ANALYZE returned the specified value.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: SNALINK

Procedure Name: chkmod

EZA5843I Using DUAL LU0 sessions

Explanation: This mode is compatible with previous versions of SNALINK.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: chkmod

EZA5844I Using SINGLE LU0 sessions

Explanation: This mode is not compatible with previous versions of SNALINK.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: chkmod

EZA5845I Got UNBIND for session

Explanation: An UNBIND request was received for a single session.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: scipfoun

EZA5847I SNA session established

Explanation: A single session was established.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: vtamopnd

EZA5848E LOSTERM received, reason *rc*

Explanation: A LOSTERM request was received.

System Action: SNALINK ends the connection.

User or Operator Response: Check your hardware or VTAM for problems.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check your hardware or VTAM for problems.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: vtamns

EZA5850I Sending SNA BIND request

Explanation: A connection is being established.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check that the SNALINK device is running.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell your VTAM administrator.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: vtamlgon

EZA5853I Processing LOGON exit

Explanation: The requested logon is being processed.

System Action: A BIND request is sent.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: To avoid this error, correct the maximum session parameter.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: To avoid this error, correct the retry interval parameter.

Source Data Set: SNALINK

Procedure Name: main

EZA5856I REPLY X TO SHUT DOWN

Explanation: SNALINK is shutting down.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Correct the values if necessary.

System Programmer Response: None.

Source Data Set: SNALINK

Procedure Name: main

EZA5866W *date time {luname}* **I 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.

User or Operator Response: The command should be reentered as several shorter commands.

System Programmer Response: None.

Source Data Set: SNALINK ASSEMBLE

Procedure Name: PROCMOD

EZA5867E *date time {luname}* **I 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.

User or Operator Response: Specify the MODIFY command with either the HALT or PKTTRACE subcommand.

System Programmer Response: None.

Source Data Set: SNALINK ASSEMBLE

Procedure Name: PROCMODF

EZA5868W *date time {luname}* **I 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.

User or Operator Response: None.

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 4 (CBD-DMO)* for an explanation of the IEE342I message.

Source Data Set: 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.

User or Operator Response: Wait for the IUCV connection between TCPIP and SNALINK to be established and reenter the command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: If multiple SNALINK LU6.2 address spaces are required, each one should have a unique name.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

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*.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LU62VTAM

Procedure Name: HEXDUMP

Chapter 7. 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See the preceding messages to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See the preceding messages in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the SNALINK LU6.2 configuration data set and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Potential problem in either VTAM or SNALINK LU6.2. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See the VTAM diagnostic messages to determine the reason for the conversation loss.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See the preceding messages in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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 3C5 reason 105.

NOTE TO DAVE HERR:

This sentence doesn't make sense — would that be ″with 3c5 reason 105″.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See the preceding messages to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See messages EZA6029E to EZA6032E in the SYSPRINT data set to determine the nature of the problem.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See the VTAM diagnostic documentation to determine the meaning of the indicated flag.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Communications Server: SNA Programmer's LU 6.2 Reference* for a description of the sense code values.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See *z/OS Communications Server: SNA Programmer's LU 6.2 Reference* for a description of the sense code values.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or 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 retry the path establishment by restarting the SNALINK LU6.2 device from within TCPIP. If the error persists, contact the IBM Software Support Center.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Locate the accompanying message EZA6060W to identify the nature of the error.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the record format of the data set, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: Enter the MODIFY command in the correct format.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the MODIFY command in the correct format.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using a valid IP address.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using a valid LU name.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using the correct syntax.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the command using a valid IP address.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Remove or correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct or remove the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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 to 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Remove one of the VTAM statements, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Add a VTAM statement to the configuration data set, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Remove one of the BUFFERS statements, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Add a BUFFERS statement to the configuration data set, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Add a DEST statement for each destination IP address in the network directly connected to the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the configuration data set, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the line in error, and restart the SNALINK LU6.2 address space.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check the indicated GENCB return code value to determine the cause of the error, and correct the problem if possible.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error or data corruption in either VTAM or SNALINK LU6.2. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program logic error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Remove the program parameters from the invocation JCL.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: Enter the required command using the correct format.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Increase the value of the REGION parameter for the SNALINK LU6.2 address space, and restart the address space.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: See message EZA6204W.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the *hlq*.PROFILE.TCPIP data set or the SNALINK LU6.2 configuration data set, and restart the address spaces.

Source Data Set: 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 retry to establish the connections every 30 seconds. The datagram is discarded.

System Action: Processing continues.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the SNALINK LU6.2 configuration data set, and restart the address space.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Probable program error. Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Correct the *hlq*.PROFILE.TCPIP data set or the SNALINK LU6.2 configuration data set, and restart the address spaces.

Source Data Set: 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.

User or Operator Response: None.

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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Check that the TCPIP initialization step involving the IEFSSNxx member of SYS1.PARMLIB was correctly performed and is in effect.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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 a 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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 EZA6279S is displayed.

System Action: The SQE server ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SQESERV

Procedure Name: await_queries

EZA6279S 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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 your 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or 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. Refer to the EZARACF sample shipped in SEZAINST for sample RACF definitions.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Verify that the SNMPD agent is available.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Increase the region size and restart the SNMP query engine.

Source Data Set: 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.

User or 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 Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SQESERV

Procedure Name: getSNMPtrap

EZA6303E recvPING: recvfrom

Explanation: The SQE server was unable to receive a SNMP ICMP packet.

System Action: Processing continues.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for information about socket errors.

Source Data Set: SQESERV

Procedure Name: create_SNMP_trap_port

EZA6305I dumping packet of *size* bytes

Explanation: The SQE server will print a packet as hexadecimal data.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell your system programmer about the error.

System Programmer Response: Determine the cause of the error by referring to the messages that follow.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reestablish the connection.

System Programmer Response: Verify that the SNMPD client is available.

Source Data Set: 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.

User or Operator Response: Contact your system programmer.

System Programmer Response: Determine if the agent is available, then reestablish the connection.

Source Data Set: 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.

User or Operator Response: Contact system programmer.

System Programmer Response: Determine if the agent is available, reestablish the connection.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Reference* for more information about socket errors.

Source Data Set: 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.

User or 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.

Source Data Set: SQEPING

Procedure Name: send_PING_packet

EZA6319W Objectid to long: obj_id

Explanation: The object identifier was too long.

System Action: MIB@DESC exits.

User or 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 Reference* for more information about the correct format of the object identifier in " "Management Information Base (MIB) Objects."

Source Data Set: 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.

User or Operator Response: Make sure that the indicated MIB description data set is loaded and in storage accessible to the host.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Respond as indicated by message EZA6320W.

System Programmer Response: Respond as indicated by message EZA6320W.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell system programmer about the error.

System Programmer Response: Increase the region size for the SNMPD program.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See Appendix for tcperror codes.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: See Appendix for tcperror codes.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Identify the SNMP agent that sent the incorrect packet, and correct any errors.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Increase the region size.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell your system programmer about the problem if it persists.

System Programmer Response: Start the SNMPQE procedure with a larger region size.

Source Data Set: 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.

User or Operator Response: Tell your 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 Reference* for more information about the correct format of the object identifier in "Management Information Base (MIB) Objects."

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Start the SNMPQE procedure with a larger region size.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Start the SNMPQE procedure with a larger region size.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Tell your 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SQEPRNV

Procedure Name: prNVparse

EZA6362W illegal protocol specification

Explanation: The SNMP query engine has received a packet with an incorrect protocol specifications. The SNMP query engine stops processing the packet.

System Action: The SNMP query engine continues.

User or Operator Response: Contact your 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Verify that the request was entered correctly.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Verify that the request was entered correctly.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: SQEPRNV

Procedure Name: prNVparse

EZA6379I null list hdr

Explanation: The SNMP query engine has receive 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Contact your 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.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 desired network.

System Action: The SNMP query engine continues processing the packet.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SQEPRNV

Procedure Name: prNVresp_copy

EZA6397I desired network: *network mask*

Explanation: The SNMP query engine has received a trap response packet. Displayed is the desired network. The desired 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 ID's.

System Action: The SNMP query engine continues processing the packet.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Verify that the request was done properly and resubmit.

System Programmer Response: None.

Source Data Set: 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 it's length. See *z/OS Communications Server: IP User's Guide and Commands*. for more information about enterprise object ID's.

System Action: The SNMP query engine continues processing the packet.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Verify that the ping request was done correctly and resubmit.

System Programmer Response: None.

Source Data Set: SQEPRNV

Procedure Name: prNVping_req

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 datset name, line number, and the object id that was not correct.

System Action: The SNMP agent continues processing the MIBDESC file.

User or Operator Response: Contact system programmer.

System Programmer Response: Correct the MIBDESC.DATA data set.

Source Data Set: MIB@DESC

Procedure Name:

EZA6424I Decoded SNMP PDU :

Explanation: SNMP has received an SNMP message.

System Action: The SNMP continues processing the SNMP message.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: If the error persists, start the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: 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.

User or Operator Response: If the error persists, restart the SNMP Query Engine with a larger region size.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or Operator Response: Tell the system programmer about the error.

System Programmer Response: Refer to *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.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MKDPI

Procedure Name: mkdpi

EZA6618W pDPI®packet: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

User or 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.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent stops processing the pDPIpacket. The SNMP agent continues.

User or 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.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPIpacket command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPItrap command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPItrap command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference* for more information about the pDPItrap command.

System Action: The SNMP agent continues processing the pDPIpacket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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.

Source Data Set: 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.

User or 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 Reference* for more information about the pDPIpacket command.

Source Data Set: 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.

User or 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 Reference* for more information about the pDPIpacket command.

Source Data Set: 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.

User or 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 Reference* for more information about the pDPIregister. command.

Source Data Set: 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.

User or 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 Reference* for more information about the mkDPIset command.

Source Data Set: 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.

User or 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 Reference* for more information about the mkDPIlist command.

Source Data Set: MKDPI

Procedure Name: dpi_set_packet

Chapter 8. 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 “?” or “Help” at the command line.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reinitiate the Telnet connection if desired.

System Programmer Response: None.

Source Data Set: TNUTMAS

Procedure Name: UtEndTelnet

EZA8247E Telnet terminated

Explanation: The Telnet connection has been ended.

System Action: The Telnet connection ends. TCPIP continues.

User or Operator Response: Reinitiate the Telnet connection if desired.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: Check for correct command syntax and try command again.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the correct data set name and resubmit the command. Refer to the *z/OS Communications Server: IP User's Guide and Commands* for more information.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP continues.

User or Operator Response: Consult your system programmer.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: The Telnet connection ends. TCPIP continues.

User or 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*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Reinitiate the Telnet connection.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: The Telnet connection ends.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Reinitiate the Telnet connection.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: The Telnet connection ends.

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

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Run a 3270 type terminal emulator or use a 3270 type display station.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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, they have been documented

with the message number for reference only. Refer to the message number in this section of the *z/OS Communications Server: IP Messages Volume 1 (EZA)* for more information. The following list provides the message description and the message number:

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
• 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

Description **Message number**

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Refer to the message number for information on the system programmer response.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP continues.

User or Operator Response: Inform the system programmer of this message.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Reinitiate the Telnet connection.

System Programmer Response: None.

Source Data Set: TNUTMAS

Procedure Name: UtGetBuffSpace

EZA8279I Connection closed

Explanation: The Telnet connection has ended.

System Action: The Telnet connection ends.

User or Operator Response: Reinitiate the Telnet connection if required.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 C, “Related protocol specifications (RFCs)” on page 587 for information about accessing RFCs.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 C, “Related protocol specifications (RFCs)” on page 587 for information about accessing RFCs.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or 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 C, “Related protocol specifications (RFCs)” on page 587 for information about accessing RFCs.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNUTMAS

Procedure Name: IacNoteArrives

EZA8308I Other Telnet char arrives

Explanation: Additional Telnet characters are being received by the server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: This message should be preceded by more specific error messages. Correct the errors indicated by the preceding messages.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNUTMAS

Procedure Name: MainLoop

EZA8313I got USERdeliveresLINE

Explanation: The user has entered a line of data at the terminal.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNUTMAS

Procedure Name: MainLoop

EZA8315I Connection closed

Explanation: The Telnet connection has ended.

System Action: The Telnet connection ends.

User or Operator Response: Reinitiate the Telnet connection if required.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: This message should be preceded by more specific error messages.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Press the ENTER key.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 desire 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNUTMAS

Procedure Name: UtFinishTelnetInit

EZA8331I Telnet

Explanation: The Telnet services have been invoked.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Submit a valid LfSet sequence in the data set to declare a new line.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Submit a valid CrSet sequence in the data set to declare a new line.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Resubmit the negotiation process.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Resubmit the Telnet command subnegtiations.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Resubmit the Telnet command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check client machine for the syntax of the command being submitted.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response:

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check the Telnet client machine for the syntax of the command being submitted.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: Resubmit the Telnet command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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* in the *z/OS Communications Server: IP Messages Volume 1 (EZA)*.

System Action: TCPIP continues

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*msgnum*.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNTOSRV

Procedure Name: SendData

EZA8372I Telnet command is

EZA8373I *command*

Explanation: Telnet issues these message while tracing is on. The current Telnet command is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Submit a correct command or type ? for help.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check the syntax and resubmit the Telnet command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Press Enter to continue.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check the syntax and resubmit the command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Enter the appropriate Telnet command.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Check the Telnet server machine for the syntax of the command being submitted.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Increase the buffer size by using the DATABUFFERPOOLSIZE statement described in the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: Proceed as instructed.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: CMMVSUB

Procedure Name: GetVirtualAddress

EZA8501I Dataset: *data_set_name*

Explanation: This message displays the name of a partitioned data set.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: See the *OS/390 MVS Programming: Authorized Assembler Services Guide* for information about the IEFDB476 service routine message displayed in this message and respond as indicated.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8521I Abnormal condition due to CSA storage shortage

Explanation: There is no CSA storage available.

System Action: TCPIP continues.

User or Operator Response: Wait a few minutes and retry. If the problem persists, contact the System Programmer.

System Programmer Response: Verify the configuration and storage definitions.

Source Data Set: *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.

User or Operator Response: If you do not want the connection to close, reestablish your connection.

System Programmer Response: None.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8523I Invalid length specified

Explanation: A parameter was of an incorrect length for the remote site.

System Action: TCPIP continues.

User or 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 your 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.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8525I Client reinitialized TCP/IP service

Explanation: A client has reinitialized TCPIP.

System Action: TCPIP restarts.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: Contact the remote site administrator for assistance.

System Programmer Response: None.

Source Data Set: *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.

User or 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.

Source Data Set: 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.

User or 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 Application Programming Interface Guide*. 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.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or Operator Response: This error should never occur. If it occurs it indicates an error with the server FTP code itself. Refer this error to your 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.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or Operator Response: Reinitialize the connection to the remote host.

System Programmer Response: Assist the operator as required.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or Operator Response: Wait a few minutes until the data set becomes available and reissue your request.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: Wait a reasonable amount of time and try accessing the data set again. If you are not successful after several tries, contact your system programmer or the IBM Software Support Center.

System Programmer Response: Assist the operator as required.

Source Data Set: *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.

User or Operator Response: Wait until the remote host is ready and reissue the connection to the remote host.

System Programmer Response: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or Operator Response: Reinitialize your connection and the application program. If the problem persists, contact your 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.

Source Data Set: *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.

User or 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 refer to 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.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8544I No TCP/IP service available

Explanation: TCPIP is not available.

User or Operator Response: Wait until TCPIP becomes available and try your request again.

System Programmer Response: None.

System Action: TCPIP halts.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8545I Not yet begun TCP/IP service

Explanation: TCPIP is not available.

System Action: TCPIP halts.

User or Operator Response: Wait until TCPIP resumes service or reinitialize TCPIP.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: Report this error to your 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.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or Operator Response: Contact the other host's administrator to determine the availability of their system.

System Programmer Response: None.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8563I Permission is denied

Explanation: The error might be caused by one of the following:

- 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.

User or Operator Response: Contact your system administrator to gain access to the specific application that you are attempting use.

System Programmer Response: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or Operator Response: If the specified request should be implemented in TCPIP, contact your 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*.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or Operator Response: Check that the IP addresses are correct in the *hlq*.PROFILE.TCPIP data set. Use the GATEWAY 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 GATEWAY statement.

System Programmer Response: None.

Source Data Set: *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.

User or 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 your 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.

Source Data Set: *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.

User or Operator Response: Wait until TCPIP resumes. If TCPIP does not resume, contact your system programmer or the IBM Software Support Center.

System Programmer Response: Increase the virtual storage for client.

Source Data Set: *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.

User or Operator Response: This error should never occur. If it does it indicates an error with the server FTP code itself. Refer this error to your 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.

Source Data Set: *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.

User or Operator Response: Wait until the problem clears, or call the remote site administrator for assistance.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: This error occurs from the client site. Wait until they restart TCPIP or call the site administrator for assistance.

System Programmer Response: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or 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: None.

Source Data Set: *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.

User or Operator Response: Resend the message using smaller packets of data or decrease the size of the message.

System Programmer Response: None.

Source Data Set: 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.

User or 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: None.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or Operator Response: Check the GATEWAY 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 GATEWAY statement see *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the operator as required.

Source Data Set: *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.

User or 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.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8595I Initiating restart, Series/1 thinks host is down

Explanation: The Sseries/1 processor is restarting the host because it has not received acknowledgments from the host.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or 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.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8599I Connection dropped by operator

Explanation: The operator intentionally dropped the connection to TCPIP.

System Action: TCPIP halts.

User or Operator Response: Reinitiate TCPIP if necessary.

System Programmer Response: None.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or 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.

Source Data Set: *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.

User or 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.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8603I Offload box restart

Explanation: The Offload device has been restarted.

System Action: TCPIP halts.

User or Operator Response: Wait until the Offload device has been completely initiated before using TCPIP.

System Programmer Response: None.

Source Data Set: *various.*

Procedure Name: *various.*

EZA8604I Offload box down

Explanation: The Offload device is not responding to TCPIP.

System Action: TCPIP halts.

User or 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.

Source Data Set: *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.

User or Operator Response: Call the IBM software Support Center for assistance.

System Programmer Response: Assist the operator as required.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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.

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

System Programmer Response: Check the client to determine why it is sending incorrect calls.

Source Data Set: 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.

User or Operator Response: Restart the connection if necessary.

System Programmer Response: Assist the user as necessary.

Source Data Set: *varies*

Procedure Name: *varies*

EZA8671I Listening

Explanation: The connection for which SayConSt was called is currently in a listening state.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *varies*

Procedure Name: *varies*

EZA8672I Nonexistent

Explanation: The connection for which SayConSt was called does not exist.

System Action: TCPIP continues.

User or Operator Response: Start the connection if necessary.

System Programmer Response: None.

Source Data Set: *varies*

Procedure Name: *varies*

EZA8673I Open

Explanation: The connection for which SayConSt was called is in an Open state.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or 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 Application Programming Interface Guide*.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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:

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or 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.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *varies*

Procedure Name: *varies*

EZA8714I Network control

Explanation: TCPIP has received an 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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.

User or 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.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 cancelling 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: *varies*

Procedure Name: *varies*

Chapter 9. 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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *varies*

Procedure Name: *varies*

EZA9002I V2: *VCMPFLG2*

Explanation: This message displays the hexadecimal value indicating the virtual machine communiction 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Application Programming Interface Guide*.

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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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* to determine whether an error has occurred and respond as indicated by the *z/OS Communications Server: IP Application Programming Interface Guide*.

Source Data Set: *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 Reference*.

System Action: The VMCF interrupt header is printed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Check the sending device to determine why it is sending undefined notifications.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *varies*

Procedure Name: *varies*

EZA9070I #*connection SayConSt(state)* **Open timeout=***time* **To read=***bytes_to_read* **Unacked=***unacknowledged_bytes* **Local= Foreign=**

Explanation: This messages indicate 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 be to 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Insert the message ID and message text in the message data set.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program retries and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Fix the message data set by removing the excess quotation marks.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Verify the name of the message data set and retry.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or Operator Response: Correct the MARGINS statement in the message file.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Increase the region size.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Refer to the originally shipped version of the specified data set and verify the name of the product.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Refer to the originally shipped version of the specified data set and verify the name of the component.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Delete 1 of the message's text and its corresponding number so there is 1 version of each.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Increase the size of the region.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Increase the size of the region.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Verify the existence of the message id and text, and retry.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Do not attempt to process any more message data sets.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or 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 Application Programming Interface Guide*.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: Falprint continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: NETSTAT ends with error message EZA0216I. TCPIP continues.

User or Operator Response: Reenter NETSTAT with the correct TCPIPJOBNAME as specified in the TCPIP.DATA dataset.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: MonQuery halts. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The TCPIP address space continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 through 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP ends.

User or 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: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: Processing continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

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

System Programmer Response: Make sure the DSCB is correctly formatted and resubmit the command.

Source Data Set: 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 Reference*.

System Action: The indicated data set is not accessed. TCPIP continues.

User or Operator Response: Recall the indicated data set. Make sure that it is properly formatted and in storage accessible to the host.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Mount the correct volume and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: Convert the data set to a direct access data set and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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 Reference*.

System Action: The parameter list is not transferred. TCPIP continues.

User or Operator Response: Make sure that the data set was specified correctly and resubmit the command.

System Programmer Response: Assist the user as necessary.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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 Application Programming Interface Guide* to determine the cause of the error.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: FTBMVSUB

Procedure Name: Exist

EZA9475I Detected trouble decoding indirected name:

EZA9477I *undecoded_name*

Explanation: The client was unable to decode an indirected 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: CMRESOL

Procedure Name: GetName

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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 C, "Related protocol specifications (RFCs)" on page 587 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 retrying a failed attempt.
- 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 C, "Related protocol specifications (RFCs)" on page 587 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: *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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: The NETSTAT program continues.

User or 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*.

Source Data Set: 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 Reference*.

System Action: The NETSTAT program halts and exits.

User or 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* .

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: Processing continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or 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.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: 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 Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMMVSUB

Procedure Name: Lqueryoff

Chapter 10. 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.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBTIINI

Appendix A. Message ranges and the components that issue them

Volume 1 ranges

Number Span	Component
EZA0519 - EZA0661	TCP/IP Utilities
EZA1450 - EZA2947	FTP
EZA2950 - EZA3172	Network Computing
EZA3950 - EZA4173	Network Database
EZA4250 - EZA4257	Portmapper
EZA4275 - EZA4361	Remote Procedure Call
EZA4375 - EZA4427	REXEC/D
EZA4726 - EZA4809	REXEC
EZA4986 - EZA5049	RSH
EZA5125 - EZA5651	SMTP
EZA5775 - EZA6230	SNALINK & SNALU6.2
EZA6275 - EZA6635	SQESERVE
EZA8200 - EZA8417	Telnet
EZA8500 - EZA9744	Common SayCalRe
EZAIN11	

Volume 2 ranges

Number Span	Component
EZB0600 - EZB0857	LPD
EZB0900 - EZB1100	LPT
EZB1200 - EZB1230	X Window
EZB2000 - EZB2498	XNX25
EZB2500 - EZB2785	Retired — do not use
EZB3000 - EZB3566	DNS
EZB3825 - EZB4198	NCPROUTE
EZB6473	Initialization
EZB8801	FFST

Volume 3 ranges

Number Span	Component
EZY0002 - EZY0058	Packet Trace
EZY0620 - EZY0974	Network Print facility
EZY1020 - EZY1035	IMS Socket
EZY1120 - EZY1130	Sockets Extended

Number Span	Component
EZY1218 - EZY1348	CICS
EZY1372 - EZY1424	REXX Socket
EZY1870 - EZY2145	MVS Platform
EZY2370 - EZY2443	MISCSRV
EZY2632 - EZY2832	FTP
EZY3720 - EZY4199	Sockets
EZY4200 - EZY4484	Telnet DBCS
EZY5398	MVS Platform
EZY6001 - EZY6077	Restartable VMCF
EZY6101 - EZY6261	FFS Writer for JES
EZYFS01 - EZYFS95 EZYFT01 - EZYFT99	FTP
EZYPR001 - EZYPR214	Network Print Facility ISPF
EZYRC01 - EZYRC24	OE REXEC
EZYRD01 - EZYRD39	OE REXECD
EZYRP35 - EZYRP72 (EZA4331 - EZA4335)	OE RPC
EZYRS01 - EZYRS66	OE RSHD
EZYTE01 - EZYTE87 EZYTO01 - EZYTO12 EZYTS02 - EZYTS18 EZYTU01 - EZYTU36 EZYTY01 - EZYTY14	OS/390 UNIX Telnet
EZYXM01 - EZYXR42	OE OSF/Motif API
EZYXU01 - EZYXU99	OE OSF/Motif User Interface Language Compiler
EZYXW01 - EZYXW69	OE x Window

Volume 4 ranges

Number Span	Component
EZZ0053 - EZZ0720	Configuration & Initialization
EZZ2351 - EZZ2389	ONETSTAT & NETSTAT
EZZ3105 - EZZ3133	OPING
EZZ3200 - EZZ3249	SNMP Subagent
EZZ3250 - EZZ3255	Operator Command
EZZ3300 - EZZ3343	OSNMP
EZZ4200 - EZZ4221	Initialization
EZZ4300 - EZZ4348	Interface Layer
EZZ4350	Initialization
EZZ4360 - EZZ4367	Storage Utilization
EZZ4820 - EZZ5027	OE ROUTED
EZZ6002 - EZZ6084	Telnet
EZZ6201 - EZZ6316	SNMP Agent
EZZ6351 - EZZ6388	OTRACERT
EZZ6452 - EZZ6705	DNS

Number Span	Component
EZZ7001 - EZZ7049	TFTP
EZZ7051 - EZZ7062	TIMED
EZZ7251 - EZZ7331	DHCP
EZZ7350 - EZZ7448	NSM
EZZ7450 - EZZ7452	Initialization
EZZ7475 - EZZ7476	Stack Configuration
EZZ7500 - EZZ7738	Sendmail & Popper
EZZ7800 - EZZ8111	OMPROUTE
EZZ8201 - EZZ8229	SLA Subagent
EZZ8252 - EZZ8260	D TCPIP,,SYSPLEX
EZZ8301 - EZZ8314	VIPA
EZ8320 - EZZ8331	HOSTNAME, DOMAINNAME, DNSDOMAINNAME, HOST
EZZ8340 - EZZ8345	Misc. Command
EZZ8392 - EZZ8420	Trap Forwarder Daemon
EZZ8431 - EZZ8451	Service Policy Agent
EZZ8453 - EZZ8460	Storage Display
EZZ8461 - EZZ8477	Sysplex Distributer
EZZ8495- EZZ8503	Intrusion Detection Services (SYSLOG)
EZZ8504 - EZZ8511	Intrusion Detection Services Statistics (TRMDSTAT)
EZZ8540	Intrusion Detection Services (SYSLOG)
EZZ8600 - EZZ8631	DCAS
EZZ8638 - EZZ8653	Intrusion Detection Services SYSLOG
EZZ8761 - EZZ8769	Intrusion Detection Services Console
EZZ8771 - EZZ8774	Policy Agent (PAGENT) Messages
EZZ8791 - EZZ9290	DNS Bind 9
EZZ9291 - EZZ9304	System Resolver
EZZ9315	Initialization
EZZ9316 - EZZ9327	Intrusion Detection Services SYSLOG
EZZ9331 - EZZ9599	DNS Bind 9
EZZ9600 - EZZ9602	SNTP
EZZ9650 - EZZ9651	VIPA
EZZ9680 - EZZ9724	DNS Bind 9
EZZ9780 - EZZ9784	Neighbor Discovery
EZZ9790 - EZZ9811	FTP
SNM000 - SNM129	SNMPLIB

Appendix B. 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:

- 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.

Refer to the *z/OS Communications Server: IP Programmer's 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. The following are 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 C. Related protocol specifications (RFCs)

This appendix lists the related protocol specifications 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.

These documents can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

where:

nnnn Is the RFC number.
TXT Is the text format.
PS Is the PostScript format.

You can see Internet drafts at http://www.ietf.org/ID.html. See "Draft RFCs" on page 594 for draft RFCs implemented in z/OS V1R4 Communications Server.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC nnnn` for text versions or a subject line of `RFC nnnn.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil`.

Many RFCs are available online. Hard copies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available using FTP from the NIC at the following Web address: http://www.rfc-editor.org/rfc.html.

Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

Many features of TCP/IP Services are based on the following RFCs:

RFC	Title and Author
768	*User Datagram Protocol* J.B. Postel
791	*Internet Protocol* J.B. Postel
792	*Internet Control Message Protocol* J.B. Postel
793	*Transmission Control Protocol* J.B. Postel
821	*Simple Mail Transfer Protocol* J.B. Postel

822 *Standard for the Format of ARPA Internet Text Messages* D. Crocker

823 *DARPA Internet Gateway* R.M. Hinden, A. Sheltzer

826 *Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware* D.C. Plummer

854 *Telnet Protocol Specification* J.B. Postel, J.K. Reynolds

855 *Telnet Option Specification* J.B. Postel, J.K. Reynolds

856 *Telnet Binary Transmission* J.B. Postel, J.K. Reynolds

857 *Telnet Echo Option* J.B. Postel, J.K. Reynolds

858 *Telnet Suppress Go Ahead Option* J.B. Postel, J.K. Reynolds

859 *Telnet Status Option* J.B. Postel, J.K. Reynolds

860 *Telnet Timing Mark Option* J.B. Postel, J.K. Reynolds

861 *Telnet Extended Options—List Option* J.B. Postel, J.K. Reynolds

862 *Echo Protocol* J.B. Postel

863 *Discard Protocol* J.B. Postel

864 *Character Generator Protocol* J.B. Postel

877 *Standard for the Transmission of IP Datagrams over Public Data Networks* J.T. Korb

885 *Telnet End of Record Option* J.B. Postel

896 *Congestion Control in IP/TCP Internetworks* J. Nagle

903 *Reverse Address Resolution Protocol* R. Finlayson, T. Mann, J.C. Mogul, M. Theimer

904 *Exterior Gateway Protocol Formal Specification* D.L. Mills

919 *Broadcasting Internet Datagrams* J.C. Mogul

922 *Broadcasting Internet Datagrams in the Presence of Subnets* J.C. Mogul

950 *Internet Standard Subnetting Procedure* J.C. Mogul, J.B. Postel

952 *DoD Internet Host Table Specification* K. Harrenstien, M.K. Stahl, E.J. Feinler

959 *File Transfer Protocol* J.B. Postel, J.K. Reynolds

974 *Mail Routing and the Domain Name System* C. Partridge

1006 *ISO Transport Service on top of the TCP Version 3* M.T.Rose, D.E. Cass

1009 *Requirements for Internet Gateways* R.T. Braden, J.B. Postel

1011 *Official Internet Protocols* J. Reynolds, J. Postel

1013 *X Window System Protocol, Version 11: Alpha Update* R.W. Scheifler

1014 *XDR: External Data Representation Standard* Sun Microsystems Incorporated

1027 *Using ARP to Implement Transparent Subnet Gateways* S. Carl-Mitchell, J.S. Quarterman

1032 *Domain Administrators Guide* M.K. Stahl

1033 *Domain Administrators Operations Guide* M. Lottor

1034 *Domain Names—Concepts and Facilities* P.V. Mockapetris

1035 *Domain Names—Implementation and Specification* P.V. Mockapetris

1042 *Standard for the Transmission of IP Datagrams over IEEE 802 Networks* J.B. Postel, J.K. Reynolds

1044 *Internet Protocol on Network System's HYPERchannel: Protocol Specification* K. Hardwick, J. Lekashman

1055 *Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP* J.L. Romkey

1057 *RPC: Remote Procedure Call Protocol Version 2 Specification* Sun Microsystems Incorporated

1058 *Routing Information Protocol* C.L. Hedrick

1060 *Assigned Numbers* J. Reynolds, J. Postel

1073 *Telnet Window Size Option* D. Waitzman

1079 *Telnet Terminal Speed Option* C.L. Hedrick

1091 *Telnet Terminal-Type Option* J. VanBokkelen

1094 *NFS: Network File System Protocol Specification* Sun Microsystems Incorporated

1096 *Telnet X Display Location Option* G. Marcy

1101 *DNS encoding of network names and other types* P.V. Mockapetris

1112 *Host Extensions for IP Multicasting* S. Deering

1118 *Hitchhikers Guide to the Internet* E. Krol

1122 *Requirements for Internet Hosts—Communication Layers* R.T. Braden

1123 *Requirements for Internet Hosts—Application and Support* R.T. Braden

1155 *Structure and Identification of Management Information for TCP/IP-Based Internets* M.T. Rose, K. McCloghrie

1156 *Management Information Base for Network Management of TCP/IP-Based Internets* K. McCloghrie, M.T. Rose

1157 *Simple Network Management Protocol (SNMP)* J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin

1158 *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* M.T. Rose

1179 *Line Printer Daemon Protocol* The Wollongong Group, L. McLaughlin III

1180 *TCP/IP Tutorial* T.J. Socolofsky, C.J. Kale

1183 *New DNS RR Definitions* C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris, (Updates RFC 1034, RFC 1035)

1184 *Telnet Linemode Option* D. Borman

1187 *Bulk Table Retrieval with the SNMP* M.T. Rose, K. McCloghrie, J.R. Davin

1188 *Proposed Standard for the Transmission of IP Datagrams over FDDI Networks* D. Katz

1191 *Path MTU Discovery* J. Mogul, S. Deering

1198 *FYI on the X Window System* R.W. Scheifler

1207 *FYI on Questions and Answers: Answers to Commonly Asked "Experienced Internet User" Questions* G.S. Malkin, A.N. Marine, J.K. Reynolds

1208 *Glossary of Networking Terms* O.J. Jacobsen, D.C. Lynch

1213 *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II* K. McCloghrie, M.T. Rose

1215 *Convention for Defining Traps for Use with the SNMP* M.T. Rose

1228 *SNMP-DPI Simple Network Management Protocol Distributed Program Interface* G.C. Carpenter, B. Wijnen

1229 *Extensions to the Generic-Interface MIB* K. McCloghrie

1230 *IEEE 802.4 Token Bus MIB* K. McCloghrie, R. Fox

1231 *IEEE 802.5 Token Ring MIB* K. McCloghrie, R. Fox, E. Decker

1236 *IP to X.121 Address Mapping for DDN* L. Morales, P. Hasse

1267 *A Border Gateway Protocol 3 (BGP-3)* K. Lougheed, Y. Rekhter

1268 *Application of the Border Gateway Protocol in the Internet* Y. Rekhter, P. Gross

1269 *Definitions of Managed Objects for the Border Gateway Protocol (Version 3)* S. Willis, J. Burruss

1270 *SNMP Communications Services* F. Kastenholz, ed.

1321 *The MD5 Message-Digest Algorithm* R. Rivest

1323 *TCP Extensions for High Performance* V. Jacobson, R. Braden, D. Borman

1325 *FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions* G.S. Malkin, A.N. Marine

1340 *Assigned Numbers* J.K. Reynolds, J.B. Postel

1348 *DNS NSAP RRs* B. Manning

1349 *Type of Service in the Internet Protocol Suite* P. Almquist

1350 *TFTP Protocol* K.R. Sollins

1351 *SNMP Administrative Model* J. Davin, J. Galvin, K. McCloghrie

1352 *SNMP Security Protocols* J. Galvin, K. McCloghrie, J. Davin

1353 *Definitions of Managed Objects for Administration of SNMP Parties* K. McCloghrie, J. Davin, J. Galvin

1354 *IP Forwarding Table MIB* F. Baker

1356 *Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode* A. Malis, D. Robinson, R. Ullmann

1363 *A Proposed Flow Specification* C. Partridge

1372 *Telnet Remote Flow Control Option* D. Borman, C. L. Hedrick

1374 *IP and ARP on HIPPI* J. Renwick, A. Nicholson

1381 *SNMP MIB Extension for X.25 LAPB* D. Throop, F. Baker

1382 *SNMP MIB Extension for the X.25 Packet Layer* D. Throop

1387 *RIP Version 2 Protocol Analysis* G. Malkin

1388 *RIP Version 2—Carrying Additional Information* G. Malkin

1389 *RIP Version 2 MIB Extension* G. Malkin

1390 *Transmission of IP and ARP over FDDI Networks* D. Katz

1393 *Traceroute Using an IP Option* G. Malkin

1397 *Default Route Advertisement In BGP2 And BGP3 Versions of the Border Gateway Protocol* D. Haskin

1398 *Definitions of Managed Objects for the Ethernet-Like Interface Types* F. Kastenholz

1416 *Telnet Authentication Option* D. Borman, ed.

1464 *Using the Domain Name System to Store Arbitrary String Attributes* R. Rosenbaum

1469 *IP Multicast over Token-Ring Local Area Networks* T. Pusateri

1535 *A Security Problem and Proposed Correction With Widely Deployed DNS Software* E. Gavron

1536 *Common DNS Implementation Errors and Suggested Fixes* A. Kumar, J. Postel, C. Neuman, P. Danzig, S.Miller

1537 *Common DNS Data File Configuration Errors* P. Beertema

1540 *IAB Official Protocol Standards* J.B. Postel

1571 *Telnet Environment Option Interoperability Issues* D. Borman

1572 *Telnet Environment Option* S. Alexander

1577 *Classical IP and ARP over ATM* M. Laubach

1583 *OSPF Version 2* J. Moy

1591 *Domain Name System Structure and Delegation* J. Postel

1592 *Simple Network Management Protocol Distributed Protocol Interface Version 2.0* B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

1594 *FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions* A.N. Marine, J. Reynolds, G.S. Malkin

1695 *Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2* M. Ahmed, K. Tesink

1706 *DNS NSAP Resource Records* B. Manning, R. Colella

1713 *Tools for DNS debugging* A. Romao

1723 *RIP Version 2—Carrying Additional Information* G. Malkin

1766 *Tags for the Identification of Languages* H. Alvestrand

1794 *DNS Support for Load Balancing* T. Brisco

1832 *XDR: External Data Representation Standard* R. Srinivasan

1850 *OSPF Version 2 Management Information Base* F. Baker, R. Coltun

1876 *A Means for Expressing Location Information in the Domain Name System* C. Davis, P. Vixie, T. Goodwin, I. Dickinson

1886 *DNS Extensions to support IP version 6* S. Thomson, C. Huitema

1901 *Introduction to Community-Based SNMPv2* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1902 *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1903 *Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1904 *Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1905 *Protocols Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1906 *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1907 *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1908 *Coexistence between Version 1 and Version 2 of the Internet-Standard Network Management Framework* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1912 *Common DNS Operational and Configuration Errors* D. Barr

1918 *Address Allocation for Private Internets* Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear

1928 *SOCKS Protocol Version 5* M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones

1939 *Post Office Protocol-Version 3* J. Myers, M. Rose

1981 *Path MTU Discovery for IP version 6* J. McCann, S. Deering, J. Mogul

1982 *Serial Number Arithmetic* R. Elz, R. Bush

1995 *Incremental Zone Transfer in DNS* M. Ohta

1996 *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)* P. Vixie

2010 *Operational Criteria for Root Name Servers* B. Manning, P. Vixie

2011 *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2* K. McCloghrie

2012 *SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2* K. McCloghrie

2013 *SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2* K. McCloghrie

2052 *A DNS RR for specifying the location of services (DNS SRV)* A. Gulbrandsen, P. Vixie

2065 *Domain Name System Security Extensions* D. Eastlake, C. Kaufman

2096 *IP Forwarding Table MIB* F. Baker

2104 *HMAC: Keyed-Hashing for Message Authentication* H. Krawczyk, M. Bellare, R. Canetti

2132 *DHCP Options and BOOTP Vendor Extensions* S. Alexander, R. Droms

2133 *Basic Socket Interface Extensions for IPv6* R. Gilligan, S. Thomson, J. Bound, W. Stevens

2137 *Secure Domain Name System Dynamic Update* D. Eastlake

2163 *Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM)* C. Allocchio

2168 *Resolution of Uniform Resource Identifiers using the Domain Name System* R. Daniel, M. Mealling

2178 *OSPF Version 2* J. Moy

2181 *Clarifications to the DNS Specification* R. Elz, R. Bush

2205 *Resource ReSerVation Protocol (RSVP) Version 1* R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin

2210 *The Use of RSVP with IETF Integrated Services* J. Wroclawski

2211 *Specification of the Controlled-Load Network Element Service* J. Wroclawski

2212 *Specification of Guaranteed Quality of Service* S. Shenker, C. Partridge, R. Guerin

2215 *General Characterization Parameters for Integrated Service Network Elements* S. Shenker, J. Wroclawski

2219 *Use of DNS Aliases for Network Services* M. Hamilton, R. Wright

2228 *FTP Security Extensions* M. Horowitz, S. Lunt

2230 *Key Exchange Delegation Record for the DNS* R. Atkinson

2233 *The Interfaces Group MIB Using SMIv2* K. McCloghrie, F. Kastenholz

2240 *A Legal Basis for Domain Name Allocation* O. Vaughn

2246 *The TLS Protocol Version 1.0* T. Dierks, C. Allen

2308 *Negative Caching of DNS Queries (DNS NCACHE)* M. Andrews

2317 *Classless IN-ADDR.ARPA delegation* H. Eidnes, G. de Groot, P. Vixie

2320 *Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2* M. Greene, J. Luciani, K. White, T. Kuo

2328 *OSPF Version 2* J. Moy

2345 *Domain Names and Company Name Retrieval* J. Klensin, T. Wolf, G. Oglesby

2352 *A Convention for Using Legal Names as Domain Names* O. Vaughn

2355 *TN3270 Enhancements* B. Kelly

2373 *IP Version 6 Addressing Architecture* R. Hinden, M. O'Dell, S. Deering

2374 *An IPv6 Aggregatable Global Unicast Address Format* R. Hinden, M. O'Dell, S. Deering

2375 *IPv6 Multicast Address Assignments* R. Hinden, S. Deering

2389 *Feature negotiation mechanism for the File Transfer Protocol* P. Hethmon, R. Elz

2428 *FTP Extensions for IPv6 and NATs* M. Allman, S. Ostermann, C. Metz

2460 *Internet Protocol, Version 6 (IPv6) S pecification* S. Deering, R. Hinden

2461 *Neighbor Discovery for IP Version 6 (IPv6)* T. Narten, E. Nordmark, W. Simpson

2462 *IPv6 Stateless Address Autoconfiguration* S. Thomson, T. Narten

2464 *Transmission of IPv6 Packets over Ethernet Networks* M. Crawford

2474 *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* K. Nichols, S. Blake, F. Baker, D. Black

2535 *Domain Name System Security Extensions* D. Eastlake

2539 *Storage of Diffie-Hellman Keys in the Domain Name System (DNS)* D. Eastlake

2553 *Basic Socket Interface Extensions for IPv6* R. Gilligan, S. Thomson, J. Bound, W. Stevens

2571 *An Architecture for Describing SNMP Management Frameworks* D. Harrington, R. Presuhn, B. Wijnen

2572 *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* J. Case, D. Harrington, R. Presuhn, B. Wijnen

2573 *SNMP Applications* D. Levi, P. Meyer, B. Stewart

2574 *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* U. Blumenthal, B. Wijnen

2575 *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* B. Wijnen, R. Presuhn, K. McCloghrie

2578 *Structure of Management Information Version 2 (SMIv2)* K. McCloghrie, D. Perkins, J. Schoenwaelder

2640 *Internationalization of the File Transfer Protocol* B. Curtin

2665 *Definitions of Managed Objects for the Ethernet-like Interface Types* J. Flick, J. Johnson

2672 *Non-Terminal DNS Name Redirection* M. Crawford

2710 *Multicast Listener Discovery (MLD) for IPv6*S. Deering, W. Fenner, B. Haberman

2711 *IPv6 Router Alert Option*C. Partridge, A. Jackson

2758 *Definitions of Managed Objects for Service Level Agreements Performance Monitoring* K. White

2845 *Secret Key Transaction Authentication for DNS (TSIG)* P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington

2874 *DNS Extensions to Support IPv6 Address Aggregation and Renumbering* M. Crawford, C. Huitema

2941 *Telnet Authentication Option* T. Ts'o, ed., J. Altman

2942 *Telnet Authentication: Kerberos Version 5* T. Ts'o

2946 *Telnet Data Encryption Option* T. Ts'o

2952 *Telnet Encryption: DES 64 bit Cipher Feedback* T. Ts'o

2953 *Telnet Encryption: DES 64 bit Output Feedback* T. Ts'o, ed.

3060 *Policy Core Information Model—Version 1 Specification* B. Moore, E. Ellesson, J. Strassner, A. Westerinen

Draft RFCs

Several areas of IPv6 implementation include elements of the following draft RFCs and are subject to change during the RFC review process.

Advanced Sockets API for IPv6
W. Richard Stevens, Matt Thomas, Erik Nordmark, Tatuya Jinmei

Basic Socket Interface Extensions for IPv6
R.E. Gilligan, S. Thomson, J. Bound, J. McCann, W. R. Stevens

Default Address Selection for IPv6
R. Draves

Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
A. Conta, S. Deering

IP Version 6 Addressing Architecture
R. Hinden, S. Deering

Appendix D. Information APARs

This appendix lists information APARs for IP and SNA documents.

Notes:

1. Information APARs contain updates to previous editions of the manuals listed below. Documents updated for V1R4 are complete except for the updates contained in the information APARs that may be issued after V1R4 documents went to press.
2. Information APARs are predefined for z/OS V1R4 Communications Server and may not contain updates.
3. Information APARs for OS/390 documents are in the document called *OS/390 DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IDDOCMST/CCONTENTS.
4. Information APARs for z/OS documents are in the document called *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com:80/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS.

Information APARs for IP documents

Table 2 lists information APARs for IP documents.

Table 2. IP information APARs

Title	z/OS CS V1R4	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8
IP API Guide	ii13255	ii12861	ii12371	ii11635
IP CICS Sockets Guide	ii13257	ii12862		ii11626
IP Configuration				ii11620 ii12068 ii12353 ii12649 ii13018
IP Configuration Guide	ii13244	ii12498 ii13087	ii12362 ii12493 ii13006	
IP Configuration Reference	ii13245	ii12499	ii12363 ii12494 ii12712	
IP Diagnosis	ii13249	ii12503	ii12366 ii12495	ii11628
IP Messages Volume 1	ii13250	ii12857 ii13229	ii12367	ii11630 13230
IP Messages Volume 2	ii13251	ii12858	ii12368	ii11631
IP Messages Volume 3	ii13252	ii12859	ii12369 12990	ii11632 ii12883
IP Messages Volume 4	ii13253	ii12860		
IP Migration	ii13242	ii12497	ii12361	ii11618

Table 2. IP information APARs (continued)

Title	z/OS CS V1R4	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8
IP Network and Application Design Guide	ii13243			
IP Network Print Facility		ii12864		ii11627
IP Programmer's Reference	ii13256	ii12505		ii11634
IP and SNA Codes	ii13254	ii12504	ii12370	ii11917
IP User's Guide			ii12365 ii13060	ii11625
IP User's Guide and Commands	ii13247	ii12501	ii12365 ii13060	ii11625
IP System Admin Guide	ii13248	ii12502		
Quick Reference	ii13246	ii12500	ii12364	

Information APARs for SNA documents

Table 3 lists information APARs for SNA documents.

Table 3. SNA information APARs

Title	z/OS CS V1R4	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8
Anynet SNA over TCP/IP				ii11922
Anynet Sockets over SNA				ii11921
CSM Guide				
IP and SNA Codes	ii13254	ii12504	ii12370	ii11917
SNA Customization	ii13240	ii12872	ii12388	ii11923
SNA Diagnosis	ii13236	ii12490 ii13034`	ii12389	ii11915
SNA Messages	ii13238	ii12491	ii12382 ii12383	ii11916
SNA Network Implementation Guide	ii13234	ii12487	ii12381	ii11911
SNA Operation	ii13237	ii12489	ii12384	ii11914
SNA Migration	ii13233	ii12486	ii12386	ii11910
SNA Programming	ii13241	ii13033	ii12385	ii11920
Quick Reference	ii13246	ii12500	ii12364	ii11913
SNA Resource Definition Reference	ii13235	ii12488	ii12380 ii12567	ii11912 ii12568
SNA Resource Definition Samples				
SNA Data Areas	ii13239	ii12492	ii12387	ii11617

Other information APARs

Table 4 on page 599 lists information APARs not related to documents.

Table 4. Non-document information APARs

Content	Number
OMPROUTE	ii12026
iQDIO	ii11220
index of recomended maintenace for VTAM	ii11220
CSM for VTAM	ii12657
CSM for TCP/IP	ii12658
AHHC, MPC, and CTC	ii01501
DLUR/DLUS for z/OS V1R2	ii12986
Enterprise Extender	ii12223
Generic resources	ii10986
HPR	ii10953
MNPS	ii10370
Performance	ii11710 ii11711 ii11712

Appendix E. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Notices

IBM may not offer all of the products, services, or features discussed in this document. 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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:

Site Counsel
IBM Corporation
P.O.Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

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 information 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 measurement 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.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 illustrates 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. 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 IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

This product includes cryptographic software written by Eric Young.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats of unlicensed books and the z/OS Licensed Product Library (LK3T-4307), which contains BookManager and PDF formats of licensed books.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
Advanced Peer-to-Peer Networking
AFP
AD/Cycle
AIX
AIX/ESA
AnyNet
APL2
AS/400
AT
BookManager
BookMaster
CBPDO
C/370
CICS
CICS/ESA
C/MVS
Common User Access
C Set ++
CT
CUA
DATABASE 2
DatagLANce
DB2
DFSMS
DFSMSdfp
DFSMShsm
DFSMS/MVS
DPI
Domino
DRDA
eNetwork
Enterprise Systems Architecture/370
ESA/390
ESCON
eServer
ES/3090
ES/9000
ES/9370
EtherStreamer
Extended Services
FAA
Micro Channel
MVS
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
MQ
Natural
NetView
Network Station
Nways
Notes
NTune
NTuneNCP
OfficeVision/MVS
OfficeVision/VM
Open Class
OpenEdition
OS/2
OS/390
OS/400
Parallel Sysplex
Personal System/2
PR/SM
PROFS
PS/2
RACF
Resource Link
Resource Measurement Facility
RETAIN
RFM
RISC System/6000
RMF
RS/6000
S/370
S/390
SAA
SecureWay
Slate
SP
SP2
SQL/DS
System/360

FFST
FFST/2
FFST/MVS
First Failure Support Technology
GDDM
Hardware Configuration Definition
IBM
IBMLink
IBMLINK
IMS
IMS/ESA
InfoPrint
Language Environment
LANStreamer
Library Reader
LPDA
MCS
System/370
System/390
SystemView
Tivoli
TURBOWAYS
UNIX System Services
Virtual Machine/Extended Architecture
VM/ESA
VM/XA
VSE/ESA
VTAM
WebSphere
XT
z/Architecture
z/OS
z/OS.e
zSeries
400
3090
3890

Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

DB2 and NetView are registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the U.S., other countries, or both.

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see http://www.intel.com/sites/corporate/tradmarx.htm .

Other company, product, and service names may be trademarks or service marks of others.

Communicating Your Comments to IBM

If you especially like or dislike anything about this document, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this document.
- If you prefer to send comments by FAX, use this number: 1-800-254-0206
- If you prefer to send comments electronically, use this network ID: usib2hpd@vnet.ibm.com

Make sure to include the following in your note:

- Title and publication number of this document
- Page number or topic to which your comment applies.

Readers' Comments — We'd Like to Hear from You

z/OS Communications Server
IP Messages:
Volume 1 (EZA)
Version 1 Release 4

Publication No. SC31-8783-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

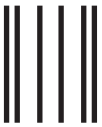
Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC31-8783-02

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Software Reengineering
Department G7IA/ Bldg 503
Research Triangle Park, NC
27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5694–A01 and 5655–G52

Printed in U.S.A.

SC31-8783-02

Spine information:

z/OS Communications Server z/OS V1R4.0 CS: IP Messages Volume 1 (EZA) Version 1 Release 4