

z/OS Communications Server



IP Messages: Volume 3 (EZY)

Version 1 Release 7

z/OS Communications Server



IP Messages: Volume 3 (EZY)

Version 1 Release 7

Note:

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 577.

Sixth Edition (September 2005)

This edition applies to Version 1 Release 7 of z/OS (5694-A01) and Version 1 Release 7 of z/OS.e (5655-G52) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You may send your comments to the following address.

International Business Machines Corporation
Attn: z/OS Communications Server Information Development
Department AKCA, Building 501
P.O. Box 12195, 3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195

You can send us comments electronically by using one of the following methods:

Fax (USA and Canada):

1+919-254-4028

Send the fax to “Attn: z/OS Communications Server Information Development”

Internet e-mail:

comsvrcf@us.ibm.com

World Wide Web:

<http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number. Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

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.

© Copyright International Business Machines Corporation 1994, 2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document	v
Who should read this document	v
How this document is organized	v
How to use this document	vi
Determining whether a publication is current	vi
How to contact IBM service	vi
Conventions and terminology used in this document	vii
The IBM TCP/IP message standards	vii
Terms and abbreviations used in this document	ix
Clarification of notes	x
Prerequisite and related information	xi
Required information	xi
Related information	xi
How to send your comments	xv
 Summary of changes	 xvii
 Chapter 1. EZY0xxxx messages	 1
 Chapter 2. EZY1xxxx messages	 61
 Chapter 3. EZY2xxxx messages	 135
 Chapter 4. EZY3xxxx messages	 181
 Chapter 5. EZY4xxxx messages	 209
 Chapter 6. EZY5xxxx messages	 265
 Chapter 7. EZY6xxxx messages	 267
 Chapter 8. EZYFxxxx messages	 303
 Chapter 9. EZYPxxxx messages	 349
 Chapter 10. EZYRxxxx messages	 373
 Chapter 11. EZYTxxxx messages	 407
 Chapter 12. EZYXxxxx messages	 447
 Appendix A. Message ranges and the components that issue them	 551
Volume 1 ranges	551
Volume 2 ranges	551
Volume 3 ranges	552
Volume 4 ranges	552
 Appendix B. Related protocol specifications (RFCs)	 555
Internet drafts	568
 Appendix C. Information APARs	 571

Information APARs for IP documents	571
Information APARs for SNA documents	572
Other information APARs	572
Appendix D. Accessibility	575
Using assistive technologies	575
Keyboard navigation of the user interface	575
z/OS information	575
Notices	577
Trademarks	585
Bibliography.	587
z/OS Communications Server information	587
z/OS Communications Server library	587
Communicating Your Comments to IBM	593

About this document

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

For information about how to set up, initialize, and customize your Transmission Control Protocol/Internet Protocol (TCP/IP) services system, 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 Guide and 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 read this document

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

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

Familiarity with TCP/IP concepts and terms is assumed.

How this document is organized

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

- Chapter 1, "EZY0xxxx messages," on page 1 contains messages in the range of EZY0002I—EZY0974I.
- Chapter 2, "EZY1xxxx messages," on page 61 contains messages in the range of EZY1020E—EZY1999E.
- Chapter 3, "EZY2xxxx messages," on page 135 contains messages in the range of EZY2000I—EZY2832E.
- Chapter 4, "EZY3xxxx messages," on page 181 contains messages in the range of EZY3720I—EZY3997E.
- Chapter 5, "EZY4xxxx messages," on page 209 contains messages in the range of EZY4000E—EZY4484I.
- Chapter 6, "EZY5xxxx messages," on page 265 contains message EZY5398E.
- Chapter 7, "EZY6xxxx messages," on page 267 contains messages in the range of EZY6001I—EZY6261W.
- Chapter 8, "EZYFxxxx messages," on page 303 contains messages in the range of EZYFS01I—EZYFT99.
- Chapter 9, "EZYPxxxx messages," on page 349 contains messages in the range of EZYPR001—EZYPR214.
- Chapter 10, "EZYRxxxx messages," on page 373 contains messages in the range of EZYRC01I—EZYRS66W.

- Chapter 11, “EZYTx_{xxxx} messages,” on page 407 contains messages in the range of EZYTE01E—EZYTY14E.
- Chapter 12, “EZYX_{xxxx} messages,” on page 447 contains messages in the range of EZYXM01W—EZYXW69I.
- Appendix A, “Message ranges and the components that issue them,” on page 551 contains a list of all message ranges and components.
- Appendix B, “Related protocol specifications (RFCs),” on page 555 lists the related protocol specifications for TCP/IP.
- Appendix C, “Information APARs,” on page 571 lists information APARs for IP and SNA documents.
- Appendix D, “Accessibility,” on page 575 describes accessibility features to help users with physical disabilities.
- “Notices” on page 577 contains notices and trademarks used in this document.
- “Bibliography” on page 587 contains descriptions of the documents in the z/OS Communications Server library.

How to use this document

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

Determining whether 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 file name (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.

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-IBM-SERV). 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 593.

Conventions and terminology used in this document

For definitions of the terms and abbreviations used in this document, you can view the latest IBM terminology at the IBM Terminology Web site.

The IBM TCP/IP message standards

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

Current numbering convention

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

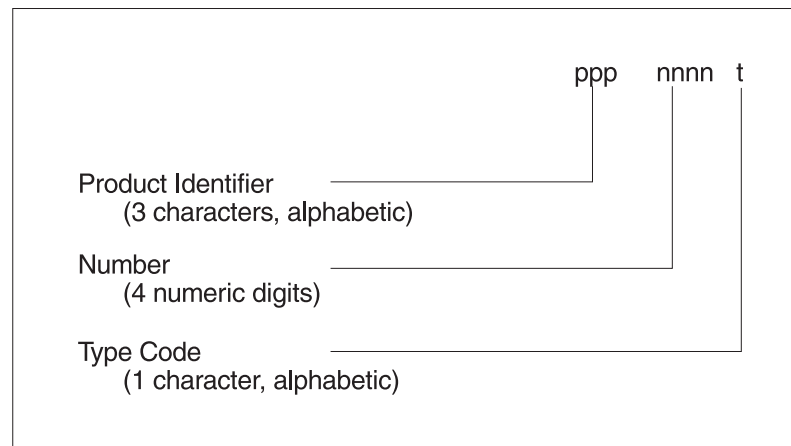


Figure 1. Sample message formats

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.

Older numbering convention

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

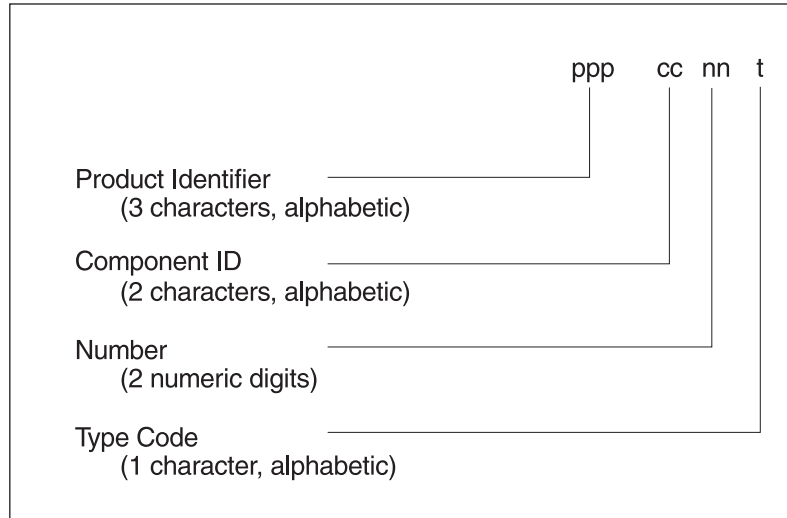


Figure 2. Sample message formats

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.

E, 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

Table 1. EZY message component identifiers (continued)

EZY component ID	Component name
RS	RSHD Client
TE	Telnet Server - Executive
TO	Telnet Server - Other
TS	Telnet Server - State
TU	Telnet Server - Utility
TY	Telnet - System
XM	OSF/Motif
XN	OSF/Motif
XO	OSF/Motif
XP	OSF/Motif
XQ	OSF/Motif
XR	OSF/Motif
XU	OSF/Motif
XW	X-Window System

Message ranges by volume

The IP Messages are distributed among volumes as follows:

Volume 1

EZA Messages

Volume 2

EZB and EZD Messages

Volume 3

EZY Messages

Volume 4

EZZ and SNM Messages

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: New Function Summary* 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.

Clarification of notes

Information traditionally qualified as **Notes** is further qualified as follows:

Note Supplemental detail

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

Guideline

Customary way to perform a procedure; stronger request than recommendation

Rule Something you must do; limitations on your actions

Restriction

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

Requirement

Dependencies, prerequisites

Result Indicates the outcome

Prerequisite and related information

z/OS Communications Server function is described in the z/OS Communications Server library. Descriptions of those documents are listed in “z/OS Communications Server information” on page 587, in the back of this document.

Required information

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

Related information

This section contains subsections on:

- “Softcopy information”
- “Other documents” on page xii
- “Redbooks” on page xii
- “Where to find related information on the Internet” on page xiii
- “Using LookAt to look up message explanations” on page xiv
- “Using IBM Health Checker for z/OS” on page xv

Softcopy information

Softcopy publications are available in the following collections:

Titles	Order Number	Description
<i>z/OS V1R7 Collection</i>	SK3T-4269	This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R7, in both BookManager and PDF formats.
<i>z/OS Software Products Collection</i>	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 <i>Getting Started with Parallel Sysplex®</i> bookshelf.
<i>z/OS V1R7 and Software Products DVD Collection</i>	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.
<i>z/OS Licensed Product Library</i>	SK3T-4307	This CD includes the licensed documents in both BookManager and PDF format.
<i>System Center Publication IBM S/390® Redbooks™ Collection</i>	SK2T-2177	This collection contains over 300 ITSO redbooks that apply to the S/390 platform and to host networking arranged into subject bookshelves.

Other documents

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 following table lists documents that might be helpful to readers.

Title	Number
<i>DNS and BIND</i> , Fourth Edition, O'Reilly and Associates, 2001	ISBN 0-596-00158-4
<i>Routing in the Internet</i> , Christian Huitema (Prentice Hall PTR, 1995)	ISBN 0-13-132192-7
<i>sendmail</i> , Bryan Costales and Eric Allman, O'Reilly and Associates, 2002	ISBN 1-56592-839-3
<i>SNA Formats</i>	GA27-3136
<i>TCP/IP Illustrated, Volume I: The Protocols</i> , W. Richard Stevens, Addison-Wesley Publishing, 1994	ISBN 0-201-63346-9
<i>TCP/IP Illustrated, Volume II: The Implementation</i> , Gary R. Wright and W. Richard Stevens, Addison-Wesley Publishing, 1995	ISBN 0-201-63354-X
<i>TCP/IP Illustrated, Volume III</i> , W. Richard Stevens, Addison-Wesley Publishing, 1995	ISBN 0-201-63495-3
<i>TCP/IP Tutorial and Technical Overview</i>	GG24-3376
<i>Understanding LDAP</i>	SG24-4986
<i>z/OS Cryptographic Service System Secure Sockets Layer Programming</i>	SC24-5901
<i>z/OS Integrated Security Services Firewall Technologies</i>	SC24-5922
<i>z/OS Integrated Security Services LDAP Client Programming</i>	SC24-5924
<i>z/OS Integrated Security Services LDAP Server Administration and Use</i>	SC24-5923
<i>z/OS JES2 Initialization and Tuning Guide</i>	SA22-7532
<i>z/OS MVS Diagnosis: Procedures</i>	GA22-7587
<i>z/OS MVS Diagnosis: Reference</i>	GA22-7588
<i>z/OS MVS Diagnosis: Tools and Service Aids</i>	GA22-7589
<i>z/OS MVS Using the Subsystem Interface</i>	SA22-7642
<i>z/OS Program Directory</i>	GI10-0670
<i>z/OS UNIX System Services Command Reference</i>	SA22-7802
<i>z/OS UNIX System Services Planning</i>	GA22-7800
<i>z/OS UNIX System Services Programming: Assembler Callable Services Reference</i>	SA22-7803
<i>z/OS UNIX System Services User's Guide</i>	SA22-7801
<i>z/OS XL C/C++ Run-Time Library Reference</i>	SA22-7821
<i>zSeries OSA-Express Customer's Guide and Reference</i>	SA22-7935

Redbooks

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

Title	Number
<i>Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 1: Base and TN3270 Configuration</i>	SG24-5227
<i>Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 2: UNIX Applications</i>	SG24-5228
<i>Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 4: Connectivity and Routing</i>	SG24-6516
<i>Communications Server for z/OS V1R2 TCP/IP Implementation Guide Volume 7: Security</i>	SG24-6840
<i>IBM Communication Controller Migration Guide</i>	SG24-6298
<i>IP Network Design Guide</i>	SG24-2580
<i>Managing OS/390® TCP/IP with SNMP</i>	SG24-5866
<i>Migrating Subarea Networks to an IP Infrastructure</i>	SG24-5957
<i>OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide: Volume 3: MVS Applications</i>	SG24-5229
<i>Secureway Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements</i>	SG24-5631
<i>SNA and TCP/IP Integration</i>	SG24-5291
<i>TCP/IP in a Sysplex</i>	SG24-5235
<i>TCP/IP Tutorial and Technical Overview</i>	GG24-3376
<i>Threadsafe Considerations for CICS</i>	SG24-6351

Where to find related information on the Internet

z/OS

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

<http://www.ibm.com/servers/eserver/zseries/zos/>

z/OS Internet Library

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

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

IBM Communications Server product

The primary home page for information about z/OS Communications Server

<http://www.software.ibm.com/network/commserver/>

IBM Communications Server product support

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

<http://www.software.ibm.com/network/commserver/support/>

IBM Systems Center publications

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

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

IBM Systems Center flashes

| Search the Technical Sales Library for Techdocs (including Flashes,
| presentations, Technotes, FAQs, white papers, Customer Support Plans,
| and Skills Transfer information)

<http://www.ibm.com/support/techdocs/atmastr.nsf>

RFCs

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

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

Internet drafts

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

<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 addresses:

USENET news groups

comp.protocols.dns.bind

BIND mailing lists

<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 (This list might not be maintained indefinitely.)

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

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.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM 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 use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM[®], VSE/ESA[™], and Clusters for AIX[®] and Linux[™]:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>.

- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX System Services).
- Your Microsoft® Windows® workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Using IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework. This book may refer to checks or messages associated with this component.

For additional information about checks and about IBM Health Checker for z/OS, see *IBM Health Checker for z/OS: User's Guide*. z/OS V1R4, V1R5, and V1R6 users can obtain the IBM Health Checker for z/OS from the z/OS Downloads page at <http://www.ibm.com/servers/eserver/zseries/zos/downloads/>.

SDSF also provides functions to simplify the management of checks. See *z/OS SDFS Operation and Customization* for additional information.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document or any other z/OS Communications Server documentation:

- Go to the z/OS contact page at:
<http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>
 There you will find the feedback page where you can enter and submit your comments.
- Send your comments by e-mail to comsvrcf@us.ibm.com. Be sure to include the name of the document, the part number of the document, the version of z/OS Communications Server, and, if applicable, the specific location of the text you are commenting on (for example, a section number, a page number or a table number).

Summary of changes

Summary of changes for SC31-8785-05 z/OS Version 1 Release 7

This document contains information previously presented in SC31-8785-04, which supports z/OS Version 1 Release 6.

The information in this document includes descriptions of support for both IPv4 and IPv6 networking protocols. Unless explicitly noted, descriptions of IP protocol support concern IPv4. IPv6 support is qualified within the text.

This document has been enabled to use the LookAt message help tool. See “Using LookAt to look up message explanations” on page xiv for information about using this tool.

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

New information

- EZY1219E
- EZY1354I —EZY1362E
- EZY1364I
- EZY1366E
- EYZFS86I

Changed information

- EZY1224I

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.

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.

Summary of changes for SC31-8785-04 z/OS Version 1 Release 6

This document contains information previously presented in SC31-8785-03, which supports z/OS Version 1 Release 5. 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

- EZYFS20I
- EZY1353E
- EZYXM06W
- EZYXM17W
- EZYXM26W
- EZYXM35W – EZYXM39W
- EZYXM43W
- EZYXM85W
- EZYXN18W – EZYXN19W
- EZYXN33W – EZYXN40W
- EZYXP43W – EZYXP44W
- EZYXP48W
- EZYXQ22W
- EZYXQ73W
- EZYXR03W
- EZYXR13W – EZYXR20W
- EZYXR51W – EZYXR53W
- EZYXR61W – EZYXR72W
- EZYXR81W – EZYXR82W
- EZYXR91W – EZYXR92W
- EZYXS01W
- EZYXS11W
- EZYXS21W
- EZYXS31W – EZYXS32W
- EZYXS41W – EZYXS46W
- EZYXS51W – EZYXS57W
- EZYXS61W – EZYXS63W
- EZYXS71W – EZYXS72W
- EZYXS81W – EZYXS86W
- EZYXS91W
- EZYXT01W
- EZYXT11W – EZYXT13W
- EZYXU79S – EZYXU82E

Changed information

- EZYFS13I
- Existing X-Window/Motif messages were updated.

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.

**Summary of changes
for SC31-8785-03
z/OS Version 1 Release 5**

This document contains information previously presented in SC31-8785-02, which supports z/OS Version 1 Release 4. 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

- EZY1349E–EZY1352E
- EZY6207R
- EZYFS16I
- EZYFS66I
- EZYFS73I–EZYFS75I
- EZYFT49I
- EZYFT77W
- EZYFT78I
- EZYFT85I
- EZYRC31I
- EZYRC33E
- EZYRD42I

Changed information

- EZY1218E
- EZY1295E
- EZY1300E–EZY1305E
- EZY1309E–EZY1311E
- EZY1315E–EZY1317E
- EZY1319E–EZY1336E
- EZY1338E
- EZY2672E
- EZYFS57I
- EZYFS79I
- EZYFT71E
- EZYRD02E

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.

Chapter 1. EZY0xxxx messages

EZY0002I **LINKNAME:** *linkname* **TRACE:** *ON/OFF*

Explanation: This is the first message displayed when the PKTTRACE LIST command is issued. The link name is displayed, as well as the current trace level. If trace is ON, then the trace options listed are currently set and active. If trace is OFF, then the trace options displayed are currently set, but no tracing will occur for the link.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0003I **Current Trace Options:**

Explanation: This message precedes the display of current trace options when the PKTTRACE LIST command is issued.

System Action: Processing continues. The trace options and trace level for the link are unchanged.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0004I **Protocol - ALL**

Explanation: All protocols will be traced for IP packets passing through the link associated with this message. This message can be displayed when the PKTTRACE LIST command is issued.

System Action: Processing continues. The trace options and trace level for the link are unchanged.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0005I **Protocol - *protocol***

Explanation: A particular well-known protocol, ICMP, TCP, or UDP, will be traced for IP packets passing through the link associated with this message. This message can be displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0006I Protocol - *protocol_number*

Explanation: A particular protocol number will be traced for IP packets passing through the link associated with this message. This message can be displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0007I Src Port - ALL

Explanation: All source ports will be traced for TCP or UDP IP packets passing through the link associated with this message. This message can be displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0008I Src Port - *source port*

Explanation: A particular source port number will be traced for TCP or UDP IP packets passing through the link associated with this message. Packets that are not TCP or UDP will not be traced. This message can be displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0009I Dest Port - ALL

Explanation: All destination ports will be traced for TCP or UDP IP packets passing through the link associated with this message. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0010I Dest Port - *destination port*

Explanation: A particular destination port number will be traced for TCP or UDP IP packets passing through the link associated with this message. Packets that are not TCP or UDP IP will not be traced. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0011I IP address - *octet1.octet2.octet3.octet4*

Explanation: This message displays the IP address. This can be used in conjunction with the subnet address mask to determine if IP packets passing through the link associated with this message will be traced. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0012I Abbrev Len - *length*

Explanation: Abbreviated IP packets will be written to the trace storage device for IP packets that are traced on the link associated with this message. The length field indicates how many bytes of the IP packet will be written, starting from the beginning of the IP header. For an ABBREV length of 0, only the packet trace header will be written. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0013I Abbrev Len - FULL

Explanation: Complete IP packets will be written to the trace storage device for IP packets that are traced on the link associated with this message. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0014I Subnet mask - *octet1.octet2.octet3.octet4*

Explanation: The subnet address mask that will be used with the IP address to determine if IP packets passing through the link associated with this message will be traced. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

EZY0015I • EZY0022W

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0015I IP address - ALL

Explanation: All IP addresses will be traced for IP packets passing through the link associated with this message. This message is displayed when the PKTTRACE LIST command is issued. The trace options and trace level for the link are unchanged.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PKTLIST

EZY0016I Clearing pkttrace options for linkname *linkname*

Explanation: The PKTTRACE CLEAR command has been issued for the link or links identified. Trace options for the link or links are set to default values and tracing is turned off.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: DOPKTTRC

EZY0017I PKTTRACE command accepted.

Explanation: The PKTTRACE command has been parsed without error. Trace options specified in the PKTTRACE command have been set. This message is displayed when a PKTTRACE command has been issued.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0022W Trace options with LIST/CLEAR ignored

Explanation: Trace options other than LINKNAME were specified in a PKTTRACE LIST or CLEAR command. The LIST or CLEAR command is executed; the other trace options are ignored.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE LIST or CLEAR command, specifying only the LINKNAME option, if required.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0023W SRCPORT/DESTPORT specified, but protocol not UDP/TCP

Explanation: The SRCPORT or DESTPORT trace option was specified in a PKTTRACE command, but the PROTOCOL trace option specified was not UDP or TCP. No IP packets will be traced for the specified link or links. The trace options are set as specified on the PKTTRACE command.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a protocol of UDP or TCP.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0024W Host identified by IP address ignored, SUBNET mask has been supplied.

Explanation: The value provided with the SUBNET parameter is used to distinguish the network and host portions of an IP address. A SUBNET mask is used to trace all packets from a particular network, it is therefore not valid to set host bits in conjunction with the SUBNET parameter. This message is generated if an IP address is supplied that has a nonzero host address. The trace options are set as specified on the PKTTRACE command. The host component of the IP address is ignored.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, ensuring all the host address bits are specified as zero in the IP address supplied with the IP trace option.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0025W No IP address specified with SUBNET

Explanation: The SUBNET trace option was specified in a PKTTRACE command, but no IP trace option was specified. The SUBNET and IP options are used in combination to determine if IP packets are to be traced. If no IP option is specified, then the default IP address mask, or that specified by a previous PKTTRACE command, will be used. The trace options are set as specified on the PKTTRACE command.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid IP address option with the SUBNET option.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0026W No options specified, no changes applied

Explanation: No options were specified with the PKTTRACE command. No changes will be made to the trace options or trace level for any links.

System Action: Processing continues.

User or Operator Response: None. Use this command to re-enable packet trace for the driver, when GTRACE has failed writing to the trace storage device. Enable or re-enable GTRACE, and then enter the PKTTRACE command with no options. Any PKTTRACE command, including one with options, could be used in this situation.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0032E Invalid option *option*.

Explanation: The specified text is not a valid option for the PKTTRACE command. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid option. The valid options are:

- ON
- OFF
- LIST
- CLEAR
- LINKNAME=
- PROT=
- IP=
- SUBNET=
- SRCPORT=
- DESTPORT=
- FULL
- ABBREV
- ABBREV=

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0033E Option *option* duplicated

Explanation: Any PKTTRACE option can be specified only once in the same PKTTRACE command. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying only one of each required option.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0034E Specify one only of ON/OFF/LIST/CLEAR

Explanation: Only one of the PKTTRACE options ON, OFF, LIST, or CLEAR can be specified in the same PKTTRACE command. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying only one of the options ON, OFF, LIST, or CLEAR.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0035E Specify one only of FULL/ABBREV

Explanation: Only one of the PKTTRACE options FULL or ABBREV can be specified in the same PKTTRACE command. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying either the FULL or ABBREV option.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0036E Invalid character specified after keyword *option*

Explanation: A required option value delimiter has not been specified. The following PKTTRACE options all require an equal sign delimiter:

- LINKNAME=
- PROT=
- IP=
- SUBNET=
- SRCPORT=
- DESTPORT=
- ABBREV=

The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying the option correctly.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0037E Invalid IP address *address*

Explanation: The supplied IP address is not valid. The address must be specified in dotted decimal notation, with exactly four octets, for example, 193.9.34.8, or the address can be specified as a single asterisk (*), to denote all IP addresses. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid IP address.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0038E No value after '=' for keyword *option*

Explanation: A required option value has not been specified. The following PKTTRACE options all require a value:

- LINKNAME=
- PROT=
- IP=
- SUBNET=
- SRCPORT=
- DESTPORT=
- ABBREV=

The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying the required value.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0039E Undefined keyword only option *option*

Explanation: The specified text is not a valid keyword-only option for the PKTTRACE command. A keyword-only option has either no value, or a default value. Any single word delimited by spaces is considered a keyword-only option because keyword value options are immediately followed by an equals sign. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid option. The valid keyword-only options are:

- ON
- OFF
- LIST
- CLEAR
- FULL
- ABBREV

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0040E Invalid protocol *protocol*

Explanation: The supplied protocol value is not valid. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid protocol value. Protocol values must be one of the following:

- A number from 0 to 255
- One of the labels ICMP, TCP, or UDP
- A single asterisk (*) used to denote all protocols

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0041E Invalid subnet address *address*

Explanation: The supplied subnet address is not valid. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid subnet address. The address must be specified in dotted decimal notation, with exactly 4 octets, for example, 193.9.34.8.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0042E Invalid abbrev value *abbrev_len*

Explanation: The supplied ABBREV value is not valid. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid ABBREV value. Valid values are integers in the range 0 to 65 535.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0043E Invalid source port *port*

Explanation: The supplied source port is not valid. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid source port value. Valid values are integers in the range 0 to 65 535, or a single asterisk (*) used to indicate all source ports.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0044E Invalid dest port *port*

Explanation: The supplied destination port is not valid. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid destination port value. Valid values are integers in the range 0 to 65 535, or a single asterisk (*) used to indicate all destination ports.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0045E Invalid linkname *linkname*

Explanation: The LINKNAME value is too long. Valid link names must be no more than 8 characters. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid link name value.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0046E IP=* invalid with SUBNET

Explanation: The SUBNET trace option cannot be specified with IP=*. An IP option of an asterisk (*) generates an implied subnet address mask (SUBNET option) of 0.0.0.0. In other words, IP=* is the equivalent of IP=0.0.0.0 SUBNET=0.0.0.0. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, and remove the SUBNET option, or specify an IP address other than *.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0047E Undefined linkname *linkname*

Explanation: The specified link name is not defined in the driver. The PKTTRACE LIST command can be used to display the defined link names for the driver. Note that SNALINK LU0 must have the IUCV connection open for the LINK before it is recognized. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a defined link name.

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0048E Undefined keyword value option *option*

Explanation: The specified text is not a valid keyword value option for the PKTTRACE command. The command is ignored. No changes are made to any trace options.

System Action: Processing continues.

User or Operator Response: Reenter the PKTTRACE command, specifying a valid option. The valid keyword value options are:

- LINKNAME=
- PROT=
- IP=
- SUBNET=
- SRCPORT=
- DESTPORT=
- ABBREV=

System Programmer Response: None.

Source Data Set: IUPKTTRC

Procedure Name: PARSEPKT

EZY0052E PKTCLEAR : Trace entry list is already clear

Explanation: The PKTCLEAR subroutine has been called to clear the linked list of trace entries, however the list contained no entries. The PKTCLEAR subroutine is called when the device driver is terminating due to a shutdown request. This is an internal error. This error will not cause system problems.

System Action: The driver continues to shut down.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center to report this internal error.

Source Data Set: IUPKTTRC

Procedure Name: PKTCLEAR

EZY0053E PKTCLEAR : FREEMAIN failed with return code *return_code*

Explanation: The PKTCLEAR subroutine has been called to clear the linked list of trace entries. The FREEMAIN system macro used to free the virtual storage for each entry has failed with the specified return code. The PKTCLEAR subroutine is called when the device driver is terminating due to a shutdown request. This might be an internal error. This error will not cause system problems.

System Action: The driver continues to shut down.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use the return code from the FREEMAIN macro to identify the problem. See *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for information about the FREEMAIN macro and

its return codes. If the error still occurs, contact the IBM Software Support Center for more information and to report this internal error.

Source Data Set: IUPKTTRC

Procedure Name: PKTCLEAR

EZY0054E PKTLINK : GETMAIN failed with return code *return_code*

Explanation: The PKTLINK subroutine has been called to allocate an entry in the linked list of trace entries. The GETMAIN macro used to allocate the virtual storage for the entry has failed with the specified return code. The PKTLINK subroutine is called during device driver initialization. Packet tracing might be disabled for some or all links.

System Action: Processing continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Define more virtual storage and try to restart the driver. Use the return code from the GETMAIN macro to identify the problem. See *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for information about the GETMAIN macro and its return codes. If the error still occurs, contact the IBM Software Support Center for more information.

Source Data Set: IUPKTTRC

Procedure Name: PKTLINK

EZY0055E PKTLINK : Linkname *linkname* already in trace entry list

Explanation: The PKTLINK subroutine has been called to allocate an entry in the linked list of trace entries. The link name specified in the subroutine call is already present in the list of trace entries. The PKTLINK subroutine is called during device driver initialization. This is an internal error. This error will not cause system problems. Packet tracing might be disabled for some or all links.

System Action: Processing continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center to report this internal error.

Source Data Set: IUPKTTRC

Procedure Name: PKTLINK

EZY0056E PKTTRACE : No linknames defined

Explanation: The PKTTRACE subroutine has been called to determine if an IP packet should be traced, however the linked list of trace entries is empty. The list of trace entries should be initialized with an entry corresponding to each link that is defined for the device driver. This is an internal error. This error will not cause system problems. Packet tracing is disabled for all links.

System Action: Processing continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center to report this internal error.

Source Data Set: IUPKTTRC

Procedure Name: PKTTRACE

EZY0057E PKTTRACE : Packet length does not include IP or UDP/TCP headers

Explanation: The PKTTRACE subroutine has been called to trace an IP packet that has been corrupted or truncated, removing headers that are required to determine if the packet is to be traced. The packet is not a valid IP packet. This could be a network problem or an internal error. This error will not cause system problems. The packet in error is not traced.

System Action: Processing continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: The formatted output that appears prior to this message indicates which connection the IP packet belongs to. Use PKTTRACE to determine if a network node is creating an IP packet that is not valid or if part of the network is damaging IP packets.

Source Data Set: IUPKTTRC

Procedure Name: PKTTRACE

EZY0058E PKTTRACE : Linkname *linkname* not in trace entry list

Explanation: The PKTTRACE subroutine has been called to determine if an IP packet should be traced, however the link that the packet was sent to or received from is not in the linked list of trace entries. The list of trace entries should be initialized with an entry corresponding to each link that is defined for the device driver. This is an internal error. This error will not cause system problems. Packet tracing might be disabled for some or all links.

System Action: Processing continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center to report this internal error.

Source Data Set: IUPKTTRC

Procedure Name: PKTTRACE

EZY0620E Incorrect function code specified for EZAPPCF Routing: Major Key = *key* Minor Key = *key*

Explanation: The VSAM function request made to module EZAPPCF is not supported. The program returns an error code to the calling module.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Obtain a dump and contact the IBM Software Support Center.

Source Data Set: EZAPPCF

Procedure Name: ERR00001

EZY0621E Storage Allocation Error in EZAPPCF Routing: Major Key = *key* Minor Key = *key*

Explanation: Insufficient virtual storage available for EZAPPCF processing.

System Action: EZAPPCF ends abnormally.

User or Operator Response: None.

System Programmer Response: Increase the region size of the address space.

Source Data Set: EZAPPCF

Procedure Name: ERR00002

EZY0622E LOAD failure for EZAPPCF Routing: Major Key = *key* Minor Key = *key*

Explanation: Module EZAPPCF is unavailable.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure module EZAPPCF is available through LINKLIST or STEPLIB specifications.

Source Data Set: EZAPPCF

Procedure Name: ERR00003

EZY0630E Specified Routing not in Routing File Routing: Major Key = *key* Minor Key = *key*

Explanation: The routing specified does not exist in the routing file. The program returns an error code to the calling module.

System Action: Processing continues.

User or Operator Response: If the routing was generated by JES, check the class, destination, and forms specified on the JES DD statement. Unless modified by an exit, a JES routing takes the form DDDD0000CFFFF000. The characters represent class, destination and forms as indicated in the following:

C Class

DDDD Destination

FFFF Forms

If the routing was generated by VTAM, check the LU name and LU type. Unless modified by an exit, a VTAM routing takes the form LUNAMExxxxxxxx. The LU name is represented in this form as follows:

LUNAMExx

 The LU name.

xxxxxxx

 Any string, but it must be specified.

If the exit is a general routing exit, specific routing exit, or input record exit, modify the routing name.

System Programmer Response: If the exit was a general routing exit, specific routing exit, or input record exit, modify the routing name and check the logic of the exit for errors.

Source Data Set: EZAPPCF

Procedure Name: ERR01001

EZY0631E Error in General Routing Exit RC=xxxx. Routing: Major Key = *key* Minor Key = *key*

Explanation: The general routing exit returned an error condition indicator in register 15. Any return codes other than zero, one, or two indicate the exit did not complete successfully.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Determine the cause of the error condition indicated by the EZAPPGPR return code. If necessary, restart the capture point after correcting the error.

Source Data Set: EZAPPCF

Procedure Name: ERR01002

EZY0632E Error in Specific Routing Exit =xxxx. Routing: Major Key = *key* Minor Key = *key*

Explanation: The specific routing exit returned with a nonzero value xxxx in register 15.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the error code to determine the cause of the problem and correct it.

Source Data Set: EZAPPCF

Procedure Name: ERR01003

EZY0633E **Number of routing records not equal to number of destinations Routing: Major Key = *key* Minor Key = *key***

Explanation: The number of routing records found for this key is not equal to the number of records specified in the type N record. The program returns an error code to the calling module.

System Action: Processing continues.

User or Operator Response: Correct the routing file.

System Programmer Response: None.

Source Data Set: EZAPPFCE

Procedure Name: ERR01004

EZY0634E **Specified Options not in Options File Routing: Major Key = *key* Minor Key = *key***

Explanation: The option specified does not exist in the options file. The program returns an error code to the calling module.

System Action: Processing continues.

User or Operator Response: Check the routing file entry for a correctly specified options entry.

System Programmer Response: If this is a general routing exit, specific routing exit, or input record exit, modify the routing name, and check the logic of the exit for errors.

Source Data Set: EZAPPFCE

Procedure Name: ERR01005

EZY0635E **VSAM Error on operation Return Code=*rc* Reason Code=*reason* DDNAME=*ddname***

Explanation: An error was encountered in a VSAM operation. The values specified in the message indicate the following:

operation

The operation that proved unsuccessful

rc

The return code

reason

The reason code

ddname

The ddname of the file

The program returns an error code to the calling module.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Refer to z/OS DFSMS Macro Instructions for Data Sets for an explanation of the reason and response codes.

Source Data Set: EZAPPFCE

Procedure Name: VSMERR50

EZY0636E **LINK Error on Specific Routing Exit.**

Explanation: The specific routing exit defined in this routing cannot be accessed. Routing: Major Key = *key* Minor Key = *key*. The program returns to the caller with an error code.

System Action: Processing continues.

User or Operator Response: Do the following:

- Note message CSV003I which gives the name of the module not found.
- Make sure the exit is correctly named in the routing file definition.
- Make sure the exit module exists in STEPLIB or JOBLIB or LPA data sets.

System Programmer Response: None.

Source Data Set: EZAPPFCE

Procedure Name: ERR01006

EZY0651E Invalid function code passed to EZAPPFCE Routing: Major Key = *key* Minor Key = *key*

Explanation: EZAPPFCE was invoked with a function code that is not valid. EZAPPFCE returns a X'0001' return code to the caller.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If the user is invoking EZAPPFCE directly, check the invoking program for a valid function code. If EZAPPFCE is being invoked by either the NPF FSS writer or NPF VTAM capture point application, contact the IBM Software Support Center.

Source Data Set: EZAPPFCE

Procedure Name: ERR00001

EZY0652E Storage allocation error in EZAPPFCE Routing: Major Key = *key* Minor Key = *key*

Explanation: An attempted GETMAIN in EZAPPFCE was unsuccessful. EZAPPFCE returns a X'0002' return code to the caller.

System Action: Processing continues.

User or Operator Response: Increase the virtual storage allocation for the address space and resubmit the job.

System Programmer Response: None.

Source Data Set: EZAPPFCE

Procedure Name: ERR00002

EZY0653E Unable to load EZAPPFCE Routing: Major Key = *key* Minor Key = *key*

Explanation: The attempt to load module EZAPPFCE was unsuccessful. EZAPPFCE returns a X'0003' return code to the caller.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Check the libraries in the STEPLIB or JOBLIB DD statement to make sure one of them contains the EZAPPFCE load module.

Source Data Set: EZAPPFCE

Procedure Name: ERR00003

EZY0654E Error loading input record exit Routing: Major Key = *key* Minor Key = *key*

Explanation: The options data specified an input record exit, but the attempt to load it was unsuccessful. EZAPPFCE returns a X'0004' return code to the caller.

System Action: Processing continues.

User or Operator Response:

1. Determine if the routing is correct.
2. Check the routing file entry for this routing. The options name specified must match an options file record.

System Programmer Response:

1. Check the JOBLIB and STEPLIB concatenations to make sure the exit module is available.
2. If any user exits modify the routing data area, check the logic of the user exits.

Source Data Set: EZAPPFCE

EZY0655E • EZY0658E

Procedure Name: ERR00004

EZY0655E **Unable to attach EZAPPLPR Routing: Major Key = *key* Minor Key = *key***

Explanation: The attempt by module EZAPPFM to ATTACH the EZAPPLPR subtask was unsuccessful. EZAPPFM returns a X'0005' return code to the caller.

System Action: For the VTAM capture point, processing continues. For the JES capture point, the program ends abnormally.

User or Operator Response: Notify the system programmer.

System Programmer Response:

1. Check the JOBLIB and STEPLIB concatenations to make sure that module EZAPPLPR is available.
2. Contact the IBM Software Support Center.

Source Data Set: EZAPPFM

EZY0656E **Error on Dynamic Allocation of data file, RC=*return code*, S99RET=*code* DSname =*name***

Explanation: Attempts to allocate a print data file were unsuccessful. The dynamic allocation return code and reason code are displayed. If the failure occurred during SMS processing, the SMS reason code (S99ERSN) displays in place of the DYNALLOC reason code (S99ERROR).

System Action: The dynamic allocation will be retried. If that attempt is also unsuccessful, an error condition will be returned to the capture point indicating a print dataset is not available.

User or Operator Response: Check for related messages that indicate insufficient DASD space or insufficient virtual storage and correct the problem indicated.

System Programmer Response: Investigate the reason for any dynamic allocation return codes contained in the message. These return and reason codes are documented in *z/OS MVS Programming: Authorized Assembler Services Guide*.

Source Data Set: EZAPPFM

Procedure Name: ERR01001

EZY0657E **OPEN failure on data file. DSname =*name***

Explanation: The print data file named could not be opened. EZAPPFM returns a X'1003' return code to the caller.

System Action: Processing continues.

User or Operator Response: Look for associated messages and correct the problems indicated.

System Programmer Response: None.

Source Data Set: EZAPPFM

Procedure Name: ERR01003

EZY0658E **CLOSE failure on data file. DSname =*name***

Explanation: The print data file named could not be closed. EZAPPFM returns a X'2001' return code to the caller.

System Action: Processing continues.

User or Operator Response: Look for associated messages and correct the problems indicated.

System Programmer Response: None.

Source Data Set: EZAPPFM

Procedure Name: ERR02001

EZY0659E PUT failure on data file. DSname =*name*

Explanation: An error occurred on a PUT to the named file. EZAPPCFCM returns a X'3002' return code to the caller.

System Action: Processing continues.

User or Operator Response: Look for associated messages and correct the problems indicated.

System Programmer Response: None.

Source Data Set: EZAPPCFCM

Procedure Name: ERR03002

EZY0660E Error on input record exit Routing: Major Key = *key* Minor Key = *key*

Explanation: A user-provided input record exit returned a nonzero return code in Register 15. EZAPPCFCM returns a X'3003' return code to the caller.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Determine the problem with the input record exit and correct it.

Source Data Set: EZAPPCFCM

Procedure Name: ERR03003

EZY0661E Record length error Routing: Major Key = *key* Minor Key = *key*

Explanation: A record length less than 0 was specified. EZAPPCFCM returns a X'3004' return code to the caller.

System Action: Processing continues.

User or Operator Response: Determine the source of the incorrect record length and correct it.

System Programmer Response: None.

Source Data Set: EZAPPCFCM

Procedure Name: ERR03004

EZY0662E Data file open for SEND DSname =*name*

Explanation: EZAPPCFCM received a request to send the data set to LPR but it is still open. Data might be lost. EZAPPCFCM returns a X'4001' return code to the caller.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If this is a user-written invocation of EZAPPCFCM, make sure the CLOSE function was invoked prior to doing the SEND. If EZAPPCFCM is invoked by the NPF FSS writer or NPF VTAM capture point, notify the IBM Software Support Center.

Source Data Set: EZAPPCFCM

Procedure Name: ERR04001

EZY0663E Data file deallocation error DSname =*name*

Explanation: An error occurred when deallocating the named data set. EZAPPCFCM returns a X'4004' return code to the caller.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Notify the IBM Software Support Center.

Source Data Set: EZAPPCFCM

EZY0664E • EZY0668E

Procedure Name: ERR04004

EZY0664E Error on Queue record creation DSname =*name*

Explanation: An error occurred on the creation of the queue record associated with the named data set. EZAPPFCCM returns a X'4004' return code to the caller.

System Action: Processing continues.

User or Operator Response: Look for associated message EZY0635E to determine the cause of the problem.

System Programmer Response: None.

Source Data Set: EZAPPFCCM

Procedure Name: ERR04004

EZY0665E Capture Point Initialization Failure

Explanation: A failure occurred at startup of the VTAM or JES capture points or during TCPIP initialization while processing in the capture points. The error is returned to the caller.

System Action: The program ends abnormally.

User or Operator Response: See the accompanying message to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZAPPFCCM

EZY0666I File Management Initialization Completed

Explanation: Network Print Facility file management initialization is complete. All parameters specified were acceptable.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFCCI

EZY0667I PRINT STARTUP USED DATASET *name*

Explanation: The startup process for Network Print Facility used the indicated data set as a source for one or more of the startup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFCCI

EZY0668E INVALID HIGH LEVEL QUALIFIER

Explanation: The high-level qualifier for the data set name is not valid. The high-level qualifier contains one of these errors:

- The qualifier is longer than 26 characters.
- The qualifier starts or ends with a period.
- The qualifier has a node name longer than 8 characters.

System Action: Processing continues.

User or Operator Response: Enter a valid high-level qualifier for the data set name and reissue the request.

System Programmer Response: None.

Source Data Set: EZAPPFCl

EZY0669E INVALID TCP/IP JOBNAME

Explanation: The job name specified is not valid; it is longer than 8 characters.

System Action: Processing continues.

User or Operator Response: Enter a valid job name and reissue the request.

System Programmer Response: None.

Source Data Set: EZAPPFCl

EZY0670E INVALID ALLOCATION PARAMETER

Explanation: The allocation parameter specified is not valid. The allocation parameter contains one of these errors:

- The type is not CYL, TRK, or a number.
- The number specified for a block size is zero or negative.
- The primary allocation is not positive or not specified.
- The secondary allocation is negative or not specified.

System Action: Processing continues.

User or Operator Response: Make sure the allocation parameters are entered correctly and reissue the request.

System Programmer Response: None.

Source Data Set: EZAPPFCl

EZY0671E INVALID THREAD SPECIFICATION, *n* WAS USED

Explanation: The specified thread value was not valid. The thread value must be greater than, or equal to one and less than, or equal to eleven. The value in *n* will be used instead.

System Action: Processing continues.

User or Operator Response: Specify a valid thread value if the default value in *n* is not acceptable.

System Programmer Response: None.

Source Data Set: EZAPPFCl

EZY0672E DEFAULT HIGH LEVEL QUALIFIER INVALID

Explanation: The specified high-level qualifier in module CMMVSYs is longer than 26 characters.

System Action: Processing continues.

User or Operator Response: Make sure the high-level qualifier in CMMVSYs is not longer than 26 characters and reissue the request.

System Programmer Response: None.

Source Data Set: EZAPPFCl

EZY0673E Insufficient space for data file.

Explanation: Network Print Facility did not have enough space to create the data file. This results in Network Print Facility issuing an x37 abend and a 4005 return code.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Increase the space allocation parameter values used for startup.

Source Data Set: EZAPPFCM

EZY0674E PUT attempted on file which is not open

Explanation: The Network Print Facility has found an error attempting to write to the QSAM print file. A return code of x'3006' is passed back to the caller.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Look for associated messages and investigate the reason why the print file may not have been opened or allocated.

Source Data Set: EZAPPFMCM

EZY0676I Startup value for *dataset value*

Explanation: This message displays the value passed to the JES writer on startup for datasets created by the JES writer.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFCI

Procedure Name: main.

EZY0677E Error on deletion of Active record

Explanation: An error was encountered while attempting to delete a record from the active file.

System Action: Processing continues.

User or Operator Response: This error will be accompanied by message EZY0635E. See that message for diagnosis information.

System Programmer Response: None.

Source Data Set: EZAPPFMCM

EZY0678E NON-ZERO RETURN CODE *xxxxxxx* FROM INITAPI CALL TO *yyyyyyyyy*

Explanation: A failure occurred at startup of the VTAM or JES capture points, or during Queue Manager initialization while attempting to initialize the SOKETS interface to the TCPIP address space. The error code from the INITAPI call (*xxxxxxx*) and the TCPIP address space name used for the call (*yyyyyyyyy*) are displayed. The error is returned to the caller.

System Action: The program ends abnormally.

User or Operator Response: Ensure that the TCPIP address space is active prior to starting the VTAM or JES capture points or the Queue Manager. An error of 00010191 indicates that the TCPIP IUCV was not installed.

System Programmer Response: Ensure that the TCPIPJOBNAME value matches the job name of the currently active TCPIP address space. If no TCPIPJOBNAME statement is found in any of the parameter data sets, the default value of TCPIP is used. For information on the extended socket return codes *xxxxxxx*, refer to *z/OS Communications Server: IP and SNA Codes*.

Source Data Set: EZAPPFCI

EZY0679I *parameter* Default value of *yyyyyyyyy* used

Explanation: This message displays the default value *yyyyyyyyy* used by the capture points or Queue manager on startup for the TCPIP global parameters. The parameter field will be either TCPIPJOBNAME or DATASETPREFIX.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFICI

EZY0680I Network Print Facility Recovery Started

Explanation: The active file contained at least one meaningful record. Each active file record that points to a closed temporary print data set will be copied into a queue file record and then deleted. All other active file records will be deleted, along with their corresponding temporary print data sets.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFCM

EZY0681I Network Print Facility Recovery Completed

Explanation: All records in the active file were processed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFCM

EZY0682E INVALID UNIT SPECIFICATION

Explanation: The UNIT name specified on the NPFUNIT statement is out of range. The range of valid UNIT values is from one to eight characters.

System Action: The error is returned to the caller.

User or Operator Response: Correct the NPFUNIT name statement in NPF.DATA. For further information about the UNIT parameter, see the *z/OS MVS JCL Reference*.

System Programmer Response: Ensure that the name specified on the NPFUNIT statement is a valid esoteric unit name, generic unit name, or unit address defined in the EDT for this system.

Source Data Set: EZAPPFICI

EZY0700E INSUFFICIENT DESTINATIONS FOR PREVIOUS ROUTING

Explanation: The previous EZAPPFL TYPE=ROUTING macro specified more than 1 destination but there are not enough EZAPPFL TYPE=NXTDEST macros to describe them.

System Action: Macro generation stops.

User or Operator Response: Either correct the number of destinations specified in the EZAPPFL TYPE=ROUTING macro or add EZAPPFL TYPE=NXTDEST macros.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .TYPE000

EZY0701E EXCESSIVE DESTINATIONS FOR PREVIOUS ROUTING

Explanation: An EZAPPFL TYPE=NXTDEST macro was encountered when it was not expected.

System Action: Macro generation stops.

User or Operator Response: Check the NDEST parameter of the previous EZAPPFL TYPE=ROUTING macro. Make sure that the number of destinations is equal to the number of EZAPPFL TYPE=NXTDEST macros following the EZAPPFL TYPE=ROUTING macro plus 1.

EZY0702S • EZY0706E

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .TYPE000

EZY0702S NO TYPE SPECIFIED IN EZAPPFL MACRO

Explanation: The EZAPPFL macro did not contain a recognizable TYPE parameter.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .TYPE010

EZY0703E NO MAJOR KEY SPECIFIED FOR ROUTING

Explanation: The EZAPPFL TYPE=ROUTING macro did not contain a MAJKEY parameter. This parameter is mandatory.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT000

EZY0704E LENGTH OF MAJOR KEY GREATER THAN 8

Explanation: The MAJKEY parameter of the EZAPPFL TYPE=ROUTING macro contained more than 8 characters. The maximum length for the MAJKEY is 8.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT005

EZY0705E NO MINOR KEY SPECIFIED FOR ROUTING

Explanation: The EZAPPFL TYPE=ROUTING macro did not contain a MINKEY parameter. This parameter is mandatory.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT010

EZY0706E LENGTH OF MINOR KEY GREATER THAN 8

Explanation: The MINKEY parameter of the EZAPPFL TYPE=ROUTING macro contained more than 8 characters. The maximum length for MINKEY is 8.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT015

**EZY0707E ROUTING KEY SEQUENCE ERROR PREVIOUS: MAJKEY=xxxxxxx MINKEY=yyyyyyy
CURRENT: MAJKEY=wwwwwww MINKEY=zzzzzzz**

Explanation: The key specified in the current EZAPPFL TYPE=ROUTING is lower in collating sequence than the previous key. The routing macros must be in ascending order based on their MAJKEY and MINKEY parameter values.

System Action: Macro generation stops.

User or Operator Response: Rearrange and reissue the routing macros.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT020

EZY0708I NUMBER OF DESTINATIONS NOT SPECIFIED, ASSUMED AS 1.

Explanation: The NDEST parameter was not specified in the EZAPPFL TYPE=ROUTING macro. The macro assumes a value of 1.

System Action: Macro generation continues.

User or Operator Response: None, if 1 is the correct value. Otherwise, specify the NDEST parameter and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT030

EZY0709E INVALID VALUE IN VTCLASS

Explanation: The VTCLASS parameter in the EZAPPFL TYPE=ROUTING macro specified a class less than 1 or greater than 64.

System Action: Macro generation stops.

User or Operator Response: Correct the class and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT060

EZY0710E OPTNAME PARAMETER REQUIRED FOR ROUTING

Explanation: The OPTNAME parameter was not specified in the EZAPPFL TYPE=ROUTING macro. This parameter is required.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT105

EZY0711E SREXIT NAME TOO LONG

Explanation: The SREXIT (specific routing exit) parameter in the EZAPPFL TYPE=ROUTING macro specified a name longer than 8 characters.

System Action: Macro generation stops.

User or Operator Response: Correct the SREXIT parameter and reissue the EZAPPFL TYPE=ROUTING macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT110

EZY0712E INAME PARAMETER REQUIRED FOR ROUTING.

Explanation: The INAME (Internet name/address) parameter was not specified in the EZAPPFL TYPE=ROUTING macro. This parameter is required.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT125

EZY0713E PNAME PARAMETER REQUIRED FOR ROUTING.

Explanation: The PNAME (printer name) parameter was not specified in the EZAPPFL TYPE=ROUTING macro. This parameter is required.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT125

EZY0714E INAME PARAMETER REQUIRED FOR NXTDEST.

Explanation: The INAME (Internet name/address) parameter was not specified in the EZAPPFL TYPE=NXTDEST macro. This parameter is required.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .NEXT010

EZY0715E PNAME PARAMETER REQUIRED FOR NXTDEST.

Explanation: The PNAME (printer name) parameter was not specified in the EZAPPFL TYPE=NXTDEST macro. This parameter is required.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT125

EZY0716E OPTNAME PARAMETER REQUIRED FOR OPTIONS

Explanation: The OPTNAME parameter was not specified in the EZAPPFL TYPE=OPTIONS macro. This parameter is required.

System Action: Macro generation stops.

User or Operator Response: Correct and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .OPTN005

EZY0717E IREXIT NAME TOO LONG

Explanation: The IREXIT (input record exit name) parameter specified in the EZAPPFL TYPE=OPTIONS was longer than 8 characters.

System Action: Macro generation stops.

User or Operator Response: Correct the name and reissue the EZAPPFL TYPE=OPTIONS macro.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .OPTN010

**EZY0718E OPTIONS KEY SEQUENCE ERROR PREVIOUS KEY=xxxxxxxxxxxxxxxxxxxxxxxxx CURRENT
KEY=yyyyyyyyyyyyyyyyyyyyyyyyyyyyyy**

Explanation: The Network Print Facility file load program has determined that the key specified in the current EZAPPFL TYPE=OPTIONS macro is lower in collating sequence than the key for the previous option name. The options macros must be specified in ascending order based on their OPTNAME parameter value.

System Action: Macro generation stops.

User or Operator Response: Rearrange the EZAPPFL TYPE=OPTIONS macros in ascending order by OPTNAME and resubmit the job.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .OPTN025

**EZY0719E VSAM ERROR MACRO=*macro* FILE=*file* RETURN=*rc* REASON=*reason* -MAJKEY:*majkey*
-MINKEY:*minkey* or -OPTNAME:*optname***

Explanation: The Network Print Facility or one of the CICS® Sockets file load programs has received a nonzero return code from VSAM services. The VSAM return and reason codes are displayed in the message following the failing VSAM macro (*macro*=OPEN, PUT, or CLOSE) and file type (*file*=ROUTING, OPTIONS, or QUEUE).

For VSAM PUTs, the error message will also include the key values, either *majkey* and *minkey* for the routine file or *optname* for the options file.

System Action: Macro generation stops.

User or Operator Response: Correct the error and resubmit the job.

System Programmer Response: Investigate the reason for the nonzero VSAM return codes displayed in the error message. These return and reason codes are documented in *z/OS DFSMS Macro Instructions for Data Sets*.

Source Data Set: EZAPPFL, EZACICD, EZACICR

Procedure Name: n/a

EZY0720S OPTIONS ENTRY PRECEEDS ROUTING ENTRY

Explanation: AN EZAPPFL TYPE=OPTIONS macro was encountered prior to an EZAPPFL TYPE=ROUTING macro. All routing entries must precede the first options entry.

System Action: Macro generation stops.

User or Operator Response: Correct the sequence of the macros and reissue them.

System Programmer Response: None.

Source Data Set: EZAPPFL

Procedure Name: .ROUT001

EZY0721W NDEST EXCEEDS 32767

Explanation: The number of destinations specified exceeds the maximum value of 32767. The value is replaced with 32767.

System Action: Processing continues.

User or Operator Response: Correct the NDEST and resubmit the job.

System Programmer Response: Assist the user in submitting an accurate NDEST parameter value. For more information on the NDEST parameter refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPFL

Procedure Name: MNOTE

EZY0722E OPTNAME PARAMETER EXCEEDS 16 CHARACTERS

Explanation: The specified OPTNAME parameter exceeds 16 characters in length. Macro generation is terminated.

System Action: Processing continues.

User or Operator Response: Correct the OPTNAME and resubmit the job.

System Programmer Response: Assist the user in submitting an accurate OPTNAME parameter value. For more information on the OPTNAME parameter refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPFL

Procedure Name: MNOTE

EZY0723E RETRY LIMIT EXCEEDS 32767

Explanation: The specified retry limit parameter exceeds the maximum value of 32767. The value is replaced with 32767.

System Action: Processing continues.

User or Operator Response: Correct the RETRY LIMIT parameter and resubmit the job.

System Programmer Response: Assist the user in submitting an accurate RETRY LIMIT parameter value. For more information on the RETRY LIMIT parameter refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPFL

Procedure Name: MNOTE

EZY0730E TIME VALUE TOO LONG

Explanation: The value specified for either RETAINT or RETRYT exceeds 8 digits. The system sets the time value to zero.

System Action: Processing continues.

User or Operator Response: Correct the time value and reissue RETAINT or RETRYT.

System Programmer Response: None.

Source Data Set: EZAPPFLT

Procedure Name: .L9

EZY0731E INVALID CHARACTER IN TIME VALUE

Explanation: The EZAPPFLT macro detected a nonnumeric character in the time value. The system sets the time value to zero.

System Action: Processing continues.

User or Operator Response: Correct the time value and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFLT

Procedure Name: .NERR1

EZY0732E INVALID DAY VALUE

Explanation: The EZAPPFLT macro detected a day value greater than 366. The system sets the time value to zero.

System Action: Processing continues.

User or Operator Response: Correct the day value and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFLT

Procedure Name: .N2

EZY0733E INVALID HOUR VALUE

Explanation: The EZAPPFLT macro detected an hour value greater than 23. The system sets the time value to zero.

System Action: Processing continues.

User or Operator Response: Correct the hour value and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFLT

Procedure Name: .NERR3

EZY0734E INVALID MINUTE VALUE

Explanation: The macro detected a minute value greater than 59. The system sets the time value to zero.

System Action: Processing continues.

User or Operator Response: Correct the minute value and reissue the macro.

System Programmer Response: None.

Source Data Set: EZAPPFLT

Procedure Name: .NERR4

EZY0750I NPF Logging Facility Started

Explanation: The Network Print Facility has started the logging activity, which will write all output to the data set specified by the EZAPPLOG DD statement.

System Action: Processing continues.

User or Operator Response: None.

EZY0751E • EZY0801I

System Programmer Response: None.

Source Data Set: EZAPPFM

EZY0751E OPEN Failure on NPF Logging File

Explanation: An error was encountered while attempting to open the data set specified by the EZAPPLOG DD statement. No logging will be done for this instance of NPF.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Verify that the file specified by the EZAPPLOG DD statement exists and is correctly specified.

Source Data Set: EZAPPFM

EZY0752E No More Space for NPF Logging File

Explanation: An error was encountered while attempting to write a record to the data set specified by the EZAPPLOG DD statement. All logging will be terminated immediately for this instance of NPF.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Increase the size of the data set specified by the EZAPPLOG DD statement and restart NPF.

Source Data Set: EZAPPFM

EZY0753E I/O Error on NPF Logging File

Explanation: The Network Print Facility has encountered an I/O error while attempting to write a record to the data set specified by the EZAPPLOG DD statement. All logging will be ended immediately for this instance of NPF.

System Action: NPF processing continues.

User or Operator Response: None.

System Programmer Response: Investigate the cause of the I/O error associated with the EZAPPCTR DD statement. Previous error messages may have been generated.

Source Data Set: EZAPPCTR

Procedure Name:

EZY0800I Starting the Network Print Facility VTAM Interface (NPFV)

Explanation: The Network Print Facility VTAM interface has been invoked.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPAAA

Procedure Name: AACMSG00

EZY0801I Terminating the Network Print Facility VTAM Interface (NPFV)

Explanation: The Network Print Facility VTAM interface is terminating.

System Action: The program ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPAAA

Procedure Name: AAAEXIT0

EZY0803E Storage allocation error

Explanation: The NPF VTAM capture point application encountered a storage allocation error.

System Action: The program ends abnormally.

User or Operator Response: Notify the system programmer.

System Programmer Response: Increase the storage allocation for the capture point.

Source Data Set: EZAPPAAA, EZAPPCHD, EZAPPCLU, EZAPPOP, EZAPPPRS, EZAPPQST, EZAPPVPP

Procedure Name: *various*

EZY0804E Timer not operational

Explanation: A timer used by the NPF VTAM capture point application is not operational. This can indicate a programming error.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: If this error results in system problems, contact the IBM Software Support Center.

Source Data Set: EZAPPCHD

Procedure Name: *various*

EZY0806E No PARM field specified

Explanation: The EXEC statement for the NPF VTAM capture point application did not contain the required PARM value field.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: NOPARMS, PARMSOK

EZY0807E PARM list ended abnormally

Explanation: MVS was unable to create a PARM list for use by the NPF VTAM capture point application. This indicates an internal error.

System Action: The application halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: *various*

Procedure Name: *various*

EZY0808E No keyword specified in the PARM field

Explanation: The EXEC statement for the NPF VTAM capture point application did not contain a required keyword before the equal sign in the PARM list.

System Action: The program ends abnormally.

EZY0809E • EZY0812E

User or Operator Response: None.

System Programmer Response: Correct the PARM list and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: GOTKEY

EZY0809E Keyword too long: *keyword*

Explanation: The EXEC statement for the NPF VTAM capture point application contained a keyword in the PARM field that was greater than 8 characters.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the PARM field and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: GOODKEYL

EZY0810E Invalid keyword: *keyword*

Explanation: The EXEC statement for the NPF VTAM capture point application contained a keyword in the PARM field that was not valid.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the PARM field and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: FINDVALU

EZY0811E No parameter specified for *keyword*

Explanation: The EXEC statement for the NPF VTAM capture point application contained a keyword in the PARM list value with no parameter specified.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the PARM field completely and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: PROCVALU

EZY0812E Invalid parameter specified for *keyword*

Explanation: The EXEC statement for the NPF VTAM capture point application contained a keyword with an incorrect parameter.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the PARM field and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: TRCLENOK

EZY0813E *parameter* **parameter too long**

Explanation: The EXEC statement for the NPF VTAM capture point application contained a keyword with that was too long.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the PARM field and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: TRCLENOK

EZY0814E **No LUCLASS specified**

Explanation: The PARM field on the EXEC statement did not specify an LUCLASS parameter. For the NPF VTAM capture point application, this parameter is required.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the PARM field and reissue the EXEC statement. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPRS

Procedure Name: PRSEXIT0

EZY0815E **No LUs were found with the specified LUCLASS**

Explanation: No match was found between the LU classes specified in the startup parameters for a VTAM application and the LU classes specified for destinations in the routing file.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Correct the startup parameter or the routing file. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPCLU

Procedure Name: CLUEXIT0

EZY0816E **MAXOPEN too big for MAXFLSTG**

Explanation: The MAXOPEN and MAXFLSTG subparameters were both specified in the JCL that starts the Network Print Facility's VTAM Capture Point application. MAXFLSTG is less than MAXOPEN multiplied by 16K (MAXOPEN * 16K) which means that MAXOPEN is not providing any meaningful limit function.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Specify a MAXFLSTG value greater than or equal to MAXOPEN * 16K or eliminate the MAXOPEN subparameter. For more information on the MAXOPEN and MAXFLSTG subparameters, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPCHD

EZY0817E Unable to load *module*

Explanation: There was a problem loading the specified module.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure that there is sufficient storage available to load the module.

Source Data Set: various

EZY0818E ATTACH of *module* failed

Explanation: The NPF VTAM capture point application was unable to attach the specified module.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure that there is sufficient storage available to load the module.

Source Data Set: EZAPPVPP

Procedure Name: VPCMSG07, ATTCHQST, ATTCHQOK

EZY0820E OPEN failed for routing file.

Explanation: The NPF VTAM capture point application was unable to open the routing file.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure the file indicated in the message exists and the DD statement is correct.

Source Data Set: EZAPPCLU

Procedure Name: ISSUERR

EZY0821E GET failed for routing file

Explanation: The NPF VTAM capture point application encountered an error trying to retrieve a routing record.

System Action: If the program was initializing when the error occurred, the program ends abnormally, otherwise, the program continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: This should be accompanied by message EZY0630E, which gives the name of the routing that proved unsuccessful. Follow the response for that message.

Source Data Set: EZAPPCLU

Procedure Name: ISSUERR

EZY0822E *name* Duplicate LU names in routing file

Explanation: The Network Print Facility VTAM capture point has found multiple VTAM routings with the same major name (*name*). There cannot be duplicate names in the routing file.

A VTAM routing is one for which a nonzero LUCLASS value has been specified. For VTAM routings, the major name is the printer's LU name. For more information on input record fields of the routing file, refer to the *z/OS Communications Server: IP Network Print Facility* manual.

System Action: The program uses the first VTAM routine with major name *name* and ignores the duplicates.

User or Operator Response: None.

System Programmer Response: Remove the duplicate name from the major name field of the routing file.

Source Data Set: EZAPPCLU

Procedure Name: BADLU

EZY0823E *xxxxxxx* - **Undetermined OPEN error, RC = rc**

Explanation: An attempt to open an LU failed for an unknown reason. The value returned in the error field of the ACB is displayed in the *rc* portion of this message.

System Action: The program marks the LU as permanently dead.

User or Operator Response: Determine what type of error occurred by looking in the *z/OS Communications Server: SNA Programming* under the OPEN macro, Completion Information section, then contact the system programmer.

System Programmer Response: For error code 36 (24 hexadecimal), correct the specified APPL definition. For all other error codes, contact the IBM Software Support Center.

Source Data Set: EZAPPCLU

Procedure Name:

EZY0824E *name* - **LU unknown to VTAM**

Explanation: The indicated LU is not known to VTAM.

System Action: The program marks the LU as retryable and will periodically attempt to open the LU.

User or Operator Response: Make sure the specified LU is defined to VTAM and activated.

System Programmer Response: Make sure the LUCLASS parameter in the JCL specifies the correct set of LU names in the routing file. Correct the startup parameter or the routing file. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPOPIN

Procedure Name: OPCMSG05

EZY0825E *name* - **LU is in the process of being opened or closed**

Explanation: A connection request has been received for the specified LU that is in the process of opening or closing.

System Action: The program marks the LU as retryable and will periodically attempt to open the LU.

User or Operator Response: Make sure the LU is not already in use by another application.

System Programmer Response: None.

Source Data Set: EZAPPOPIN

Procedure Name: OPCMSG06

EZY0826E *LU name* - **LU already in use by another application**

Explanation: A connection request has been received for the specified LU which is already active.

System Action: The system marks the LU as retryable and will periodically attempt to open the LU.

User or Operator Response: None.

System Programmer Response: Correct the startup parameter or the routing file. Also, make sure the specified LU is not already in use by another application. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPOPIN

Procedure Name: OPCMSG07

EZY0827E *name* - LU was improperly defined in VTAM definition deck

Explanation: The specified LU name that is assigned to the NPF VTAM capture point application has attributes that are inconsistent with this use.

System Action: The program ignores the LU.

User or Operator Response: Make sure the LU is not already in use.

System Programmer Response: Correct the startup parameter or the routing file. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPOP

Procedure Name: OPCMSG08

EZY0828E *number* LUs unknown to VTAM

Explanation: The indicated number of LUs are not known to VTAM.

System Action: The program marks the LUs as retryable and will periodically attempt to open them.

User or Operator Response: Make sure the LUs are defined to VTAM and activated.

System Programmer Response: Make sure the LUCLASS parameter in the JCL specifies the correct set of LU names in the routing file. Correct the startup parameter or the routing file. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPOP

Procedure Name: OPCMSG05

EZY0829E *number* LUs in the process of being opened or closed

Explanation: The indicated number of LUs are in the process of opening or closing.

System Action: The program marks the LUs as retryable and will periodically attempt to open them.

User or Operator Response: Make sure the LUs are not already in use by another application.

System Programmer Response: None.

Source Data Set: EZAPPOP

Procedure Name: OPCMSG06

EZY0830E *number* LUs already in use by another application

Explanation: A connection request has been received for the indicated number of LUs, which are already active.

System Action: The system marks the LUs as retryable and will periodically attempt to open them.

User or Operator Response: None.

System Programmer Response: Correct the startup parameter or the routing file. Also, make sure the specified LU is not already in use by another application. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPOP

Procedure Name: OPCMSG07

EZY0831E *number* LUs improperly defined in VTAM definition deck

Explanation: The indicated number of LUs are assigned to the NPF VTAM capture point application and have attributes that are inconsistent with this use.

System Action: The program ignores the LUs.

User or Operator Response: Make sure the LUs are not already in use.

System Programmer Response: Correct the startup parameter or the routing file. For more information on the EXEC statement, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPOPNI

Procedure Name: OPCMSG08

EZY0832E SETLOGON failed for name

Explanation: An attempt was made to issue a SETLOGON START for an open ACB.

System Action: The program closes the ACB and will periodically retry the OPEN ACB and SETLOGON operations.

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

System Programmer Response: Make sure enough storage is available for VTAM.

Source Data Set: EZAPPOPNI

Procedure Name: OPCMSG09

EZY0833E VTAM is currently inactive

Explanation: The VTAM support on the system is not active.

System Action: The program continues to run and will periodically retry the VTAM OPEN ACB operation.

User or Operator Response: Make sure VTAM is active before starting Network Print Facility for VTAM.

System Programmer Response: None.

Source Data Set: EZAPPOPNI

Procedure Name: OPCMSG03

EZY0834E VTAM is shutting down

Explanation: The VTAM support on the system is being shut down.

System Action: The program continues to run and will periodically retry the VTAM OPEN ACB operation.

User or Operator Response: Reactivate VTAM or end Network Print Facility for VTAM.

System Programmer Response: None.

Source Data Set: EZAPPOPNI

Procedure Name: OPCMSG04

EZY0835E No VTAM exists on the system

Explanation: The system contains no VTAM support.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure VTAM is installed before running Network Print Facility for VTAM.

Source Data Set: EZAPPOPNI

Procedure Name: OPCMSG02

EZY0837I Enter NPFV application operator command

Explanation: This message prompts for an operator command. The possible commands and their usage can be found in the *z/OS Communications Server: IP Network Print Facility*.

System Action: Processing continues.

User or Operator Response: Refer to the *z/OS Communications Server: IP Network Print Facility* for information on NPFV application operator commands and their usage.

EZY0838E • EZY0842I

System Programmer Response: None.

Source Data Set: EZAPPA, EZAPPCHD

Procedure Name: AAIWTOR

EZY0838E Invalid reply specified

Explanation: The reply to message EZY0837I is not correct.

System Action: The system prompts for another reply.

User or Operator Response: Re-enter the correct reply.

System Programmer Response: None.

Source Data Set: EZAPPCHD

Procedure Name: CHCMSG02

EZY0839E LU specified not in the routing file

Explanation: The LU specified on the LUNAME operator reply was not found in the routing file.

System Action: Processing continues.

User or Operator Response: Make sure the correct LUNAME is specified and reenter the request.

System Programmer Response: Update the routing file if necessary.

Source Data Set: EZAPPCLU

Procedure Name:

EZY0840E LU specified already open

Explanation: The requested LU was already open.

System Action: Processing continues.

User or Operator Response: Make sure that the proper LU was requested.

System Programmer Response: None.

Source Data Set: EZAPPCLU

Procedure Name: DOSRCH1

EZY0841E Routing file record not specified as NPF VTAM record

Explanation: The user requested a new LU be opened by the NPF VTAM capture point application. The one requested is not marked as a VTAM record.

System Action: Processing continues.

User or Operator Response: Make sure the correct LU name was entered.

System Programmer Response: Make sure the routing file has been configured correctly.

Source Data Set: EZAPPCLU

Procedure Name: BADLU

EZY0842I LUNAME ADD succeeded

Explanation: The LU was successfully created.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPCHD

Procedure Name: OPNLUXIT

EZY0843E LUNAME ADD failed

Explanation: There was an error while attempting to create a Logical Unit.

System Action: Processing continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Determine if the appropriate parameters were specified on the EXEC statement for the creation of an LU.

Source Data Set: EZAPPCHD

Procedure Name: CREATFLD

EZY0844I LUNAME DEL succeeded

Explanation: A connection has been successfully deleted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPCHD

Procedure Name: n/a

EZY0845E LUNAME DEL failed - LU does not exist

Explanation: A connection could not be deleted because the specified connection does not exist.

System Action: Processing continues.

User or Operator Response: Make sure the connection was properly specified.

System Programmer Response: None.

Source Data Set: EZAPPCHD

Procedure Name: n/a

EZY0846E LUNAME DEL failed - LU still OPEN

Explanation: A connection could not be deleted. The logical unit is still open.

System Action: Processing continues.

User or Operator Response: Make sure that the LU is properly specified and not in use by another application.

System Programmer Response: None.

Source Data Set: EZAPPCHD

Procedure Name: n/a

EZY0847I Table reload complete

Explanation: The end-of-file rules table or the default page format table has been reloaded after the RELOAD command has been issued. For more information, refer to the *z/OS Communications Server: IP Network Print Facility* manual.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

EZY0848E • EZY0851E

Source Data Set: EZAPPCHD

Procedure Name: FREECOM

EZY0848E **LOAD failed for new table**

Explanation: There was an error loading the new end-of-file rules table or the default page format table. Refer to *z/OS Communications Server: IP Network Print Facility* for more information.

System Action: Processing continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Make sure that the table exists and is in the proper library, and that sufficient storage is available to support the table.

Source Data Set: EZAPPCHD

Procedure Name: LOADERR1

EZY0849E **Format invalid for new table**

Explanation: The new end-of-file rules table or the default page format table format was unrecognized. For information on the page formatting macros, refer to *z/OS Communications Server: IP Network Print Facility*.

System Action: Processing continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Correct the table format and recompile and link edit the corrected table.

Source Data Set: EZAPPCHD

Procedure Name: LOADERR2

EZY0850I **Command accepted**

Explanation: The requested command has been accepted by the Network Print Facility. This message is issued in response to QUIT, QUIT FORCE, or KILL commands in reply to message EZY0837I, or STOP in reply to message EZY0960I. The command has been accepted as valid and shutdown of the NPF jobs has begun. There may be a significant delay between the issuing of this message and the completion of the shutdown process.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPCHD, EZAPPQSB

EZY0851E **INVALID INVOCATION OF EZAPPDPE**

Explanation: EZAPPDPE is a macro used by EZAPPDPE. This message indicates a logic error in EZAPPDPE. The EZAPPDPE macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Call the IBM Software Support Center to report the problem.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPDPE

EZY0852E INVALID CHARACTER IN *field* VALUE

Explanation: Only numbers between 0 and 9 can be specified in the value field. The EZAPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the value in the specified field of the EZAPDPF macro. Use only numerics between 0 and 9 in the value field.

Source Data Set: EZAPPPFT

Procedure Name: EZAPDPPE

EZY0854E PAGE FORMAT NAME IS REQUIRED

Explanation: Each page format you specify with the EZAPDPF macro in the EZAPPPFT module must have a name. The EZAPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Specify a name for this page format table entry on the EZAPDPF macro. For the proper syntax of the macro, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPFT

Procedure Name: EZAPDPF

EZY0855E PAGE FORMAT NAME TOO LONG

Explanation: The page format name you specified with the EZAPDPF macro in the EZAPPPFT module is longer than eight characters. For more information on the EZAPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the EZAPDPF macro so the entry name is the correct length. For proper syntax of the macro, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPPFT

Procedure Name: EZAPDPF

EZY0856E MAX PRES. POSITION GREATER THAN 255

Explanation: The maximum presentation position (MPP) is an optional keyword representing the line length. This can be an integer from 0 to 255. A value of 0 uses the printer's default of 80 columns. For more information on this keyword, refer to *z/OS Communications Server: IP Network Print Facility*. The macro EZAPDPF is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the MPP parameter in the EZAPDPF macro to a number from 0 to 255.

Source Data Set: EZAPPPFT

Procedure Name: EZAPDPF

EZY0857E LEFT MARGIN CANNOT EXCEED MPP

Explanation: The maximum presentation position (MPP) represents the line length. The left margin keyword (LM), which identifies where the left margin starts, cannot have a value greater than the value for MPP. A value of 0 defaults to column 1. For more information on the keywords of the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the LM parameter in the EZAPPPDPF macro to have a value between 0 and the value for MPP.

Source Data Set: EZAPPPDPF

Procedure Name: EZAPPPDPF

EZY0858E RIGHT MARGIN < LM OR > MPP

Explanation: The keyword representing where the right margin starts has a value that is either less than the value for left margin (LM) or it is greater than the value for the maximum presentation position (MPP). A value of 0 is also permitted and sets the right margin equal to the maximum presentation position (MPP). For more information on the keywords for the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the right margin (RM) value to be 0 or less than the value for MPP and greater than the value for LM.

Source Data Set: EZAPPPDPF

Procedure Name: EZAPPPDPF

EZY0859E HT_n < LM OR > RM

Explanation: The nth value specified for the horizontal tabs (HT=(...)) is not valid because it is either less than the left margin or greater than the right margin. A value of 0 is also permitted and is ignored. For more information about the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the horizontal tabs (HT) value to be 0 or greater than the LM value and less than the RM value.

Source Data Set: EZAPPPDPF

Procedure Name: EZAPPPDPF

EZY0860E MAX PRES. LINE GREATER THAN 255

Explanation: The maximum presentation line (MPL) is an optional keyword representing the page length in lines. This value can be an integer from 0 to 255. A value of 0 uses the printer default of 1 line. For more information on this keyword of the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the maximum presentation line (MPL) value to a number between 0 and 255.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0861E TOP MARGIN CANNOT EXCEED MPL

Explanation: The top margin (TM) keyword specifies what line number the top margin starts on. This value cannot be greater than the value for the maximum page length (MPL). A value of 0 sets the top margin to line 1. For more information on this keyword of the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the value for the TM keyword in the EZAPPPDPF macro to a number between 0 and the value for MPL.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0862E BOTTOM MARGIN < TM OR > MPL

Explanation: The value for the bottom margin (BM) cannot be less than the top margin (TM) or greater than the page length (MPL). A value of 0 is also permitted and sets the bottom margin equal to the page length. For more information on this keyword of the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the BM value in the EZAPPPDPF macro to be 0 or greater than TM and less than MPL.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0863E VT_n < TM OR > BM

Explanation: The nth value specified for vertical tabs (VT=(...)) was not valid. The value must be less than the value for BM (bottom margin) or greater than the value for TM (top margin). A value of 0 is also permitted and is ignored. For more information about the EZAPPPDPF macro, refer to *z/OS Communications Server: IP Network Print Facility* manual. The EZAPPPDPF macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the value for the vertical tabs (VT) to be 0 or greater than the value for TM and less than the value for BM.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0864I MAX PRES. POSITION DEFAULTED TO 80

Explanation: The optional maximum presentation position (MPP) keyword was not included in the EZAPPPDPF macro syntax. The MPP will default to a value of 80.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0865I LEFT MARGIN DEFAULTED TO 1

Explanation: The left margin (LM) was not specified in the EZAPPPDPF macro. The LM will default to 1. To change the left margin, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPPDPF macro.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0866I RIGHT MARGIN DEFAULTED TO MPP

Explanation: The right margin (RM) was not specified in the EZAPPPDPF macro. The RM will default to the value of MPP. For information on how to change the RM, refer to *z/OS Communications Server: IP Network Print Facility*.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0867I HT_n DEFAULTED TO 0

Explanation: The nth horizontal tab parameter was omitted from the list specified with the HT=(*ht1,ht2,...*) keyword on the EZAPPPDPF macro. The nth horizontal tab will default to a value of 0. For information on the HT keyword, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPPDPF macro.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0868I MAX PRES. LINE DEFAULTED TO 1

Explanation: The maximum presentation line (MPL) keyword was not found in the EZAPPPDPF macro. MPL will default to a value of 1. The MPL represents the page length in lines.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0869I TOP MARGIN DEFAULTED TO 1

Explanation: The optional top margin (TM) keyword was not found in the EZAPPPDPF macro. TM will default to a value of 1. For more information on this keyword, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPPDPF macro.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0870I BOTTOM MARGIN DEFAULTED TO MPL

Explanation: The bottom margin (BM) was not found in the EZAPPPDPF macro. BM will default to a value equal to the value of MPL. For more information on this keyword, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPPDPF macro.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0871I VT_n DEFAULTED TO 0

Explanation: The nth vertical tab parameter was omitted from the list specified with the VT=(*vt1,vt2,...*) keyword on the EZAPPPDPF macro. The nth vertical tab will default to a value of 0. For more information on this keyword, refer to *z/OS Communications Server: IP Network Print Facility*.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0872W DEFAULT PAGE FORMAT NOT GENERATED

Explanation: No entry name was specified, or one or more of the following had a non-numeric value as its parameter:

- MPP
- LM
- RM
- HT
- MPL
- TM
- BM
- VT

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Verify that all the keywords in the EZAPPPDPF macro have correct values. Also, verify that the format has an entry name. Refer to *z/OS Communications Server: IP Network Print Facility* for more information on the keywords.

Source Data Set: EZAPPPFT

Procedure Name: EZAPPPDPF

EZY0873E STB VALUE *xxx* IS INVALID

Explanation: The only valid settings for the STB operand (Suppress Trailing Blanks) are YES and NO. The EZAPDPDF macro is not expanded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the value in the STB field of the EZAPDPDF macro.

Source Data Set: EZAPPPFT

EZY0874E TYPE=END MUST BE LAST MACRO

Explanation: If the TYPE parameter in the EZAPPEFM macro is END, the macro must mark the end of the table. For information on other possibilities of the TYPE parameter, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Make sure that the EZAPPEFM TYPE=END macro call is the last statement in your EZAPPEFT source deck.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0875E TYPE PARAMETER IS REQUIRED

Explanation: The TYPE keyword is required for identifying the function of the EZAPPEFM macro. The 4 possible values are:

- SYSDFLT
- ENTRY
- PLU
- END

For information on these values, refer to *z/OS Communications Server: IP Network Print Facility* manual. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: When coding the EZAPPEFM macro, make sure to include the TYPE keyword and an appropriate value.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0876E TYPE PARAMETER INVALID

Explanation: When specifying the TYPE value when coding the EZAPPEFM macro, use one the following values:

- SYSDFLT
- ENTRY
- PLU
- END

For more information on these values, refer to *z/OS Communications Server: IP Network Print Facility* manual. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the value for the TYPE keyword of the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0877E TYPE=SYSDFLT MUST BE THE FIRST MACRO

Explanation: The SYSDFLT parameter was present for the TYPE keyword when the EZAPPEFM macro was coded. It must be the first EZAPPEFM macro and it must be the only EZAPPEFM macro in the module with the SYSDFLT value. For information on the values of the TYPE keyword, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the value of the TYPE value in the EZAPPEFM macro or make this the first EZAPPEFM macro in the module.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0878E MUST FOLLOW VALID TYPE=ENTRY MACRO

Explanation: Before coding the TYPE=PLU keyword in the EZAPPEFM macro, there must be a TYPE=ENTRY keyword in a previous EZAPPEFM macro. For more information on the TYPE keyword, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Add an EZAPPEFM macro with the TYPE=ENTRY keyword, before the macro with TYPE=PLU.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0879E TABLE ENTRY NAME IS REQUIRED

Explanation: When coding the TYPE=ENTRY keyword in the EZAPPEFM macro, you must specify the name for the end-of-file rules table entry as the label on the macro. For information on the EZAPPEFM macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Include name of the end-of-file rules table entry as the label on the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0880E TABLE ENTRY NAME TOO LONG

Explanation: When coding the EZAPPEFM TYPE=ENTRY macro, the macro label (name of the table entry) must be between 1 and 8 alphanumeric characters long. For information about the EZAPPEFM macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro ends abnormally.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the macro label to contain 8 or less alphanumeric characters.

Source Data Set: EZAPPEFT

EZY0881E • EZY0884E

Procedure Name: EZAPPEFM

EZY0881E NAME FOUND WHEN NOT TYPE=ENTRY

Explanation: A label on the EZAPPEFM macro is only permitted when the TYPE is ENTRY. Refer to *z/OS Communications Server: IP Network Print Facility* for more information on the macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the EZAPPEFM macro adding the TYPE=ENTRY keyword or deleting the macro label.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0882E PLUNAME IS REQUIRED

Explanation: While coding the EZAPPEFM macro, the TYPE=PLU keyword was found and there is no value for the PLUNAME. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Add the PLUNAME keyword and value to the macro. For information on the macro syntax and the PLUNAME keyword, refer to *z/OS Communications Server: IP Network Print Facility*.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0883E PLUNAME FOUND WHEN NOT TYPE=PLU

Explanation: When the TYPE value is not PLU, there must not be a keyword PLUNAME. For information the EZAPPEFM macro, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Remove the PLUNAME keyword or change the TYPE value to PLU.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0884E PLUNAME TOO LONG

Explanation: The value for the PLUNAME keyword of the macro syntax is too long. The maximum number of alphanumeric characters that a PLUNAME value can be is 8. For more information, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the PLUNAME to be less than or equal to 8 alphanumeric characters.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0885E PLUNAME HAS NO SIGNIFICANT CHARS

Explanation: The PLUNAME value specified for the PLUNAME keyword contained no characters other than * or ?. For naming conventions for the PLUNAME value, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Change the value specified by the PLUNAME keyword in the EZAPPEFM macro syntax to contain at least 1 significant character.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0886E PLUNAME HAS IMBEDDED BLANK OR *

Explanation: An asterisk or a blank was found in the middle of the PLUNAME value. The asterisk can be used as a wildcard at the beginning or the end of a PLUNAME value, but not in the middle. Blank spaces are not permitted in the middle of a PLUNAME value either. For naming conventions for the PLUNAME value, refer to the *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the PLUNAME value in the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0887E PLUNAME HAS MIXED * AND ?

Explanation: Both an asterisk and a question mark were found in the PLUNAME value. Refer to the *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro for information on naming conventions for the PLUNAME value. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the PLUNAME value in the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0888E EOFn= NOT ALLOWED WITH EOF=

Explanation: Both the EOFn and the EOF keywords were found in the syntax for the EZAPPEFM macro. Refer to the *z/OS Communications Server: IP Network Print Facility* manual for information on the end-of-file rules. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Delete the appropriate keyword from the macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0889E EOF SPEC REQUIRED

Explanation: The TYPE value in the EZAPPEFM macro was either SYSDFLT or PLU. With these values, an EOF or EOFn keyword is required. For information on these values, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Add an appropriate EOF or EOFn value to the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0890E EOF SPEC NOT ALLOWED ON TYPE=END

Explanation: The EOF keywords are not allowed when the TYPE value is END. For more information on the EOF specifications, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Remove the EOF keyword from the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0891E *parameter* PARAMETER UNRECOGNIZED

Explanation: The value specified for the EOF or EOFn keyword on the EZAPPEFM macro is incorrect. The possible values are:

- EB
- EC
- ES
- STRING
- TIMER

For more information on these values, refer to *z/OS Communications Server: IP Network Print Facility*. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Replace the EOF or EOFn value with the appropriate value.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0892E EOFn IDLE/BUSY TIMES REQUIRED

Explanation: When entering the value for the EOFn=(TIMER,*idleint*,*busyint*) keyword, the TIMER value must be followed by an idle interval and a busy interval, each specified in seconds. For information on these parameters, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Add the *idleint* and *busyint* parameters to the value of the EOF or EOFn keyword.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0893E EOFn STRING KEEP / DEL REQUIRED

Explanation: If the STRING value is used when coding the end-of-file rules (EOF) keyword for the EZAPPEFM macro, it must be accompanied by a KEEP or DEL parameter. For information on these additions, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Add the KEEP or DEL parameter to the STRING value in the coding of the EOF keyword of the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0894E EOFn STRING VALUE REQUIRED

Explanation: When coding the STRING value for the EOF keyword in the EZAPPEFM macro, a parameter must be included. The parameter can either be a character or hexadecimal value that does not exceed 56 characters. For more information, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the macro syntax to contain a parameter for the STRING value of the EOF keyword.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0895E EOFn TIMER NOT ALLOWED WITH LU1

Explanation: The TIMER value can be used with EOF0 or EOF3 and cannot be used with EOF or EOF1. For more information, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Specify separate rules for EOF0, EOF1, and EOF3. Use a rule other than TIMER for the EOF1 keyword of the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0896E EOFn STRING DATATYPE NOT C OR X

Explanation: When coding the STRING value of the EOF keyword of the EZAPPEFM macro, the parameter must be a hexadecimal string (X) or a character string (C). Refer to *z/OS Communications Server: IP Network Print Facility* for more information. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the value of the string in the EZAPPEFM macro.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0897E EOFn QUOTE REQUIRED AFTER C OR X

Explanation: Following the C or X in the STRING parameter, there must be a quotation mark before the parameter. For more information on the parameters of the STRING value, refer to *z/OS Communications Server: IP Network Print Facility* under the EZAPPEFM macro. The EZAPPEFM macro is not expanded.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Correct the macro by adding a single quotation mark before the string and after the X or C in the EZAPPEFM macro. Remember to terminate the string with a single quotation mark also.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0898E MACRO DID NOT EXPAND

Explanation: The macro syntax was incorrect. This message is displayed with another error message indicating what was not correct.

System Action: The assembler continues.

User or Operator Response: None.

System Programmer Response: Use the message preceding this one to determine the error and correct the macro syntax.

Source Data Set: EZAPPEFT

Procedure Name: EZAPPEFM

EZY0910I fssid printerid SYSOUT DYNALLOC/OPEN DONE

Explanation: The creation and opening of the print data set is complete.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: OPNDDDD1

EZY0911I WILL RETRY FOR DYNALLOC

Explanation: The first attempt to dynamically allocate a data set was not successful.

System Action: The program continues and again attempts to dynamically allocate a data set.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: OPNDDDD1

EZY0912I fssid printerid JOB jobid jobname IS HELD IN JES QUEUE DUE TO NO VALID DESTINATION

Explanation: Network Print Facility determined there is no valid routing record or options record for the print file. Network Print Facility does not know where to send the print file.

System Action: Processing continues.

User or Operator Response: Define a valid routing record or options record, and reissue the print request.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: OPNDDDD1

EZY0915I *fssid printerid* **JOB jobid jobname HELD IN JES QUEUE DUE TO NON-ZERO GETREC RC**

Explanation: The Network Print Facility has received an error indication from JES during GETREC processing. No further processing of the output data set can be completed.

System Action: The spool file on which the JES GETREC error occurred is changed to a HELD status in the JES queue. Processing continues with the next available print dataset.

User or Operator Response: Ensure that the job creating the output has completed successfully. If the output is not printable, it may need to be purged from the JES queue.

System Programmer Response: Investigate the reason for the nonzero GETREC return code. A previous error message may have been generated, listing a GETREC error condition such as GLRLGE or GLRIOE. These codes are documented in the JES GETREC Processing section of *z/OS MVS Using the Functional Subsystem Interface*.

Source Data Set: EZAPPFA

EZY0920I *fssid printerid* **SYSOUT PUT/CLOSE DONE**

Explanation: The creation of a print data set is complete and the close has been performed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: CLSDDDD1

EZY0921I *fssid printerid* **SYSOUT SEND DONE**

Explanation: The transmission of the print data set via LPR is complete.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: CLSDDDD1

EZY0922I *fssid printerid* **SYSOUT RELEASE DONE**

Explanation: The JES writer has performed a termination function. This indicates the writer is ending and has successfully closed all files and detached subtasks related to Network Print Facility.

System Action: The program ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: PFSADD1

EZY0930I *fssid printerid* **RESET TO FILE START REQUESTED**

Explanation: An input record exit has requested the JES writer to reset the file to the beginning for reprocessing. The system restarts the JES output file.

System Action: Processing continues.

EZY0931E • EZY0934E

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: SFSDEXIT

EZY0931E GETMAIN FOR DEVICE DRIVER SDCB FAILED

Explanation: The JES device driver is unable to allocate its work space due to lack of virtual storage.

System Action: The system ends the JES device driver (writer).

User or Operator Response: Increase the virtual storage assigned to the writer and restart.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: SFSADD1

EZY0932E *ERROR* RC=*rc* FROM FCM. FSA IS TERMINATING..

Explanation: The JES writer has received a recoverable error response from EZAPPFDD.

System Action: The system ends the JES device driver (writer).

User or Operator Response: Look for an accompanying message to provide details of the error. Respond based on the accompanying message.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: SFSDEXIT

EZY0933E MINOR ERROR RC=*rc* FROM FCM IS IGNORED

Explanation: The JES writer has received a recoverable error response from EZAPPFDD such as an incorrect routing.

System Action: The system continues and processes the next JES output.

User or Operator Response: Look for an accompanying message to provide details of the error. Respond based on the accompanying message.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: SFSDEXIT

EZY0934E *ERROR* CAN NOT LOAD EZAPPFDD

Explanation: The JES writer failed in attempting to LOAD EZAPPFDD.

System Action: The system ends the JES device driver (writer).

User or Operator Response: Check the JOBLIB/STEPLIB assignments to make sure that the EZAPPFDD module is accessible. If so, look for indications of a load being unsuccessful from MVS. Correct the error and retry.

System Programmer Response: None.

Source Data Set: EZAPPFDD

Procedure Name: SFSDEXIT

EZY0939W *fssid printerid* **WARNING! Truncating print data record length. Exceeds FSS maximum of 32756.**

Explanation: The largest print data set record length for SPIN=DATASET has been exceeded. In order to prevent the FSS from ABENDING, the print data has been truncated.

System Action: The NPF print data set is created with the maximum record length of 32756 and processing continues.

User or Operator Response: Determine that no actual data was lost when the truncation occurred. If automatic truncation is not desired, the user can modify the DCB parameters on the DD statement within the job that creates the print data set. The combination that necessitated truncation of data is RECFM=U and BLKSIZE greater than 32756.

Although it may affect other processing in the user's NPF environment, the problem can also be avoided by modifying the FSS to SPIN=GROUP. Make sure that SPIN=GROUP is acceptable before modifying this parameter.

System Programmer Response: None.

Source Data Set: EZAPPFDD

EZY0951E **VSAM macro failure on** *macro* **RETURN CODE =** *rc* **REASON CODE =** *reason*

Explanation: VSAM has encountered an error processing the Queue file.
MMMMMMMM

The name of the VSAM macro that was unsuccessful.

RRRR The return code provided by VSAM (in)

CCCC The reason code provided by VSAM

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Refer to *z/OS DFSMS Macro Instructions for Data Sets* for an explanation of the reason and response codes.

Source Data Set: EZAPPQSA

Procedure Name: QSIMSGPV

EZY0952E **NPF Queue Manager: Incorrect parameter/no parameter specified.**

Explanation: The parameters specified in the PARM field of the JCL EXEC statement are either missing or incorrect.

System Action: The Queue Manager program ends abnormally.

User or Operator Response: Correct the parameters and reissue the EXEC statement. You must specify at least one parameter at startup, the time interval between queue scans. Specify this value in hours, minutes, and seconds format *hhmmss*.

System Programmer Response: None.

Source Data Set: EZAPPQSA

Procedure Name: QSIMSG02

EZY0953E **ATTACH for EZAPPQSB failed.**

Explanation: The ATTACH for the subtask running EZAPPQSB (timer subtask) was unsuccessful. Either the module could not be found or there is insufficient virtual storage to load it.

System Action: The programs ends abnormally.

User or Operator Response: Check the STEPLIB and JOBLIB data sets or the link list for the presence of this module. If it is found, increase the virtual storage and retry the job.

System Programmer Response: None.

Source Data Set: EZAPPQSA

Procedure Name: QSIMSG03

EZY0954E Unable to load EZAPPFCA.

Explanation: The LOAD for module EZAPPFCA was unsuccessful. Either the module could not be found or there is insufficient virtual storage to load it.

System Action: The program ends abnormally.

User or Operator Response: Check the STEPLIB and JOBLIB data sets or the link list for the presence of this module. If it is found, increase the virtual storage and retry the job.

System Programmer Response: None.

Source Data Set: EZAPPQSA

Procedure Name: QSIMSG04

EZY0955E Insufficient storage for program

Explanation: The virtual storage allocation for EZAPPQSA is not sufficient to run the program.

System Action: The program ends abnormally.

User or Operator Response: Increase the region size and rerun the program.

System Programmer Response: None.

Source Data Set: EZAPPQSA

Procedure Name: QSI00010

EZY0956E DYNALLOC macro failure on ALLOCATION/DEALLOCATION RETURN CODE = rc

Explanation: An error (nonzero value in) was returned by the DYNALLOC macro (SVC 99). rc is the return code.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Refer to *z/OS MVS Programming: Authorized Assembler Services Guide* for an explanation of the return code.

Source Data Set: EZAPPQSA

Procedure Name: QSIMSGPD

EZY0957I Routing route key operation File filename At time on day day year year

Explanation: Trace output. The routing identified by *route key* has completed operation *operation* using file *file* at the time stated.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQSA

Procedure Name: QSITMMSG

EZY0958E Initialization error in Queue Manager

Explanation: An error occurred during initialization in the queue manger. The accompanying message describes the specific error.

System Action: Queue manager stops processing.

User or Operator Response: See the accompanying message to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZAPPQSA

EZY0959E CLOSE failure on Queue File

Explanation: The Network Print Facility Queue Manager encountered an error trying to close the QUEUE file while ending. The error message is issued and the Queue Manager continues with ending process.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Look for associated messages and investigate the reason why the QUEUE file may not have been successfully closed.

Source Data Set: EZAPPQSA

EZY0960I NPF Queue Manager: Enter STOP or new time value

Explanation: The operator can reply to this message to change the scan interval or stop the queue manager.

System Action: The system changes the operation as specified.

User or Operator Response: To change the scan interval, reply with a value of hours, minutes, and seconds in the form *hhmmss*. To stop the queue manager, reply with the word *STOP*. To start a trace of the NPF Queue Manager to the system console, reply with the word *TRACE*. To stop tracing the NPF Queue Manager to the system console, reply *NOTRACE*. To change the maximum printer queue depth, reply with a new value from 0-99999 in the form *QDEPTH=nnnnnn*. Specifying *QDEPTH=0* will remove any previously set *QDEPTH* limits and set *DEPTHWTO* to a value of *N*. To request that a WTP be issued when *QDEPTH* is reached, reply with *DEPTHWTO=Y*. *QDEPTH* must not be a nonzero value before *DEPTHWTO* can be turned on. To request that a WTO no longer be issued when *QDEPTH* is reached, reply with *DEPTHWTO=N*.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0961I NPF Queue Manager: Incorrect Reply.

Explanation: The reply to message EZY0960I is not correct.

System Action: The system reissues message EZY0960I.

User or Operator Response: Respond correctly to the new message.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0962E Unable to Attach EZAPPLPR

Explanation: The queue manager was unable to attach module EZAPPLPR.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Make sure that the module EZAPPLPR is in the proper load library and that enough storage is available to load the module. Increase the region size if necessary.

Source Data Set: EZAPPQSA

EZY0963E EZAPPLPR Task Abended Completion Code = cc

Explanation: The task EZAPPLPR has ended abnormally. The reason is indicated by the MVS system completion code displayed in this message. The queue manager may retain the print data set for reprocessing according to user options in effect at the time of the error.

System Action: The task ends abnormally.

User or Operator Response: If the error persists, stop queue manager and notify the system programmer.

System Programmer Response: Use the MVS completion code displayed in this message and the MVS system

documentation to determine the cause of the error and respond as indicated.

Source Data Set: EZAPPQSA

EZY0964I Queue manager trace turned {on | off}

Explanation: The TRACE for the Queue Manager has been activated (on) or deactivated (off). This message indicates completion of a TRACE or NOTRACE command entered in reply to message EZY0960I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0965I Queue manager interval changed

Explanation: The interval for Queue Manager scans has been changed in a reply to message EZY0960I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0966I Queue Manager DEPTHWTO turned off

Explanation: No WTO will be issued when the current QDEPTH limit is reached for any printer.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0967I Queue Manager DEPTHWTO turned on

Explanation: A WTO will be issued when the current QDEPTH limit has been reached for any printer.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0968I Queue Manager QDEPTH changed

Explanation: QDEPTH has been changed to the new value specified.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQSB

EZY0970I *yy/ddd hh:mm:ss REC data_set_name majorname minorname #recs #dests*

Explanation: An entire print file has been received by NPF and queued under the indicated data set name. Fields in the message are as follows:

yy/ddd hh:mm:ss

Julian date and time.

majorname

For VTAM, the SLU name. For JES, the jobname.

minorname

For VTAM, the PLU name. For JES, the username.

#recs Number of records in the file.

#dests Number of destinations to which the file will be sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPCTR

EZY0971I *yy/ddd hh:mm:ss SND data_set_name printqueue host RC00000 rcdaction datasetaction*

Explanation: An attempt to send the print file for the specified data set to LPD for printing has been successful. Fields in the message are as follows:

yy/ddd hh:mm:ss

Julian date and time.

printqueue

The print queue name (truncated to 10 bytes, if necessary).

host The TCPIP host name or IP address (truncated to 30 bytes, if necessary).

RC00000

Indicates an LPR return code = 0.

Note: RC50xx is a return code presented from LPR. For additional documentation, invoke LPR with the trace option specified against the dataset *data_set_name* referred to in the message.

rcdaction

The following actions can occur:

- QUE (queue) indicates that a queue file record has been created/updated to allow future retries of the send operation by the NPF queue manager.
- DEL (delete) indicates that the queue file record has been deleted.
- QER (queue error) indicates an error occurred while writing a queue file record. The data has been kept to allow future retries of the send operation through manual invocation of LPR. The NPF queue manager is unable to take any further action.

datasetaction

RET (retain) or DEL (delete) indicates whether the corresponding print data set has been kept or deleted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPCTR

EZY0972E *yy/dddd hh:mm:ss SND data_set_name printqueue host errcode rcdaction datasetaction*

Explanation: An attempt to send the print file for the specified data set to LPD for printing has ended with an error or abend indication. Fields in the message are as follows:

yy/dddd hh:mm:ss

Julian date and time.

printqueue

The print queue name (truncated to 10 bytes, if necessary).

host The TCPIP host name or IP address (truncated to 30 bytes, if necessary).

errcode The following types of errors can occur:

- RCrrrrr where rrrrr = LPR bad return code in decimal.

Note: RC50xx is a return code presented from LPR. For additional documentation, invoke LPR with the trace option specified against the dataset *data_set_name* referred to in the message.

- AB00aaa where aaa = system abend code from LPR in hex.
- ATTFAIL indicates that an ATTACH of EZAPPLPR failed.

rcdaction

The following actions can occur:

- QUE (queue) indicates that a queue file record has been created/updated to allow future retries of the send operation by the NPF queue manager.
- DEL (delete) indicates that the queue file record has been deleted.
- QER (queue error) indicates an error occurred while writing a queue file record. The data has been kept to allow future retries of the send operation through manual invocation of LPR. The NPF queue manager is unable to take any further action.

datasetaction

RET (retain) or DEL (delete) indicates whether the corresponding print data set has been kept or deleted.

System Action: Processing continues.

User or Operator Response: Turn on LPR trace to get more detailed diagnostic messages.

System Programmer Response: None.

Source Data Set: EZAPPCTR

EZY0973I *yy/dddd hh:mm:ss PRG data_set_name printqueue host rcdstate rcdaction datasetaction*

Explanation: The NPF queue manager has purged a record from the queue file for one of the following reasons:

- RETAIN(S) retain time expired after a successful transmission.
- RETAIN(U) retain time expired after all retries were exhausted for an unsuccessful transmission.
- A record was marked for deletion by the NPF panel operator.
- The end-of-set (state=X) record was found to be the only remaining record for a multi-destination routing.

The NPF queue manager may also have deleted the print data set specified by the purged queue file record:

- For a single-destination routing, the print data set is deleted when the one queue file record for that routing is purged.
- For a multi-destination routing, the print data set must be retained until all the queue file records for that routing can be processed. Only when the last of these queue file records is purged is the print data set deleted.

Fields in the message are as follows:

yy/dddd hh:mm:ss

Julian date and time.

printqueue

The print queue name (truncated to 10 bytes, if necessary), or blank (for rcdstate=X only).

host The TCPIP host name or IP address (truncated to 30 bytes, if necessary), or blank (for rcdstate=X only).

rcdstate

The state of the queue file record at the time it was purged:

- T - successfully transmitted
- U or R- unsuccessful
- D - deletion requested by operator
- X - end-of-set record for a multi-destination routing

rcaction

DEL indicates that the queue file record has been deleted.

datasetaction

RET (retain) or DEL (delete) indicates whether the corresponding print data set has been kept or deleted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPCTR

EZY0974I **Queue Manager reached QDEPTH for: *printq* at *host***

Explanation: The QDEPTH limit specified has been reached for the printer displayed in the message.

System Action: Processing continues.

User or Operator Response: Investigate why the printer queue reached its limit. Possible reasons include:

- The printer is not ready to print.
- An unusually large number of print data sets has been sent to the printer within a relatively short period of time.

System Programmer Response: None.

Source Data Set: EZAPPQSA

Chapter 2. EZY1xxxx messages

EZY1020E *hh:mm:ss.th* **Line** *linenum*: *error_description* - '*token*'

Explanation: An error was encountered while processing a statement in the configuration data set. The line number of the statement in error is given by *linenum*. The message includes a description of the error and *token* identifies the specific text that is in error. Explanations of the errors described by *error_description* are:

Required valid statement absent

The end of the data set was reached and a valid instance of a required statement was not found. If the statement was present but contained an error, it will have already been flagged by a message.

The missing statement type is identified by *token*. The required statement types are: TCPIP, LISTENER, and TRANSACTION.

Required parameter missing or incorrect

A required parameter was not specified or was specified incorrectly. The name of the required parameter is identified by *token*.

The statement is ignored.

Prior statement conflict

The parameters specified on the TRANSACTION statement encountered on line *linenum* conflict with parameters specified on a previous TRANSACTION statement. The NAME parameter on both statements, identified by *token*, was the same, but the other parameters were different.

The transaction definition is removed from the set of transactions that can be started by the Listener.

Redeclaration of prior statement

The TRANSACTION statement encountered on line *linenum*, with NAME *token*, is the same as a previous TRANSACTION statement.

The current statement is ignored.

No parameter keyword

The parameter *token* has been specified without an identifying keyword and therefore cannot be recognized. The parameter is ignored.

No parameter value

The parameter *token* does not specify a value when one is required.

The parameter is ignored.

Unknown statement keyword

The statement type given by *token* is unknown. Valid statement types are: TCPIP, LISTENER, and TRANSACTION.

The statement is ignored.

Unknown parameter keyword

The parameter *token* specifies an unknown parameter keyword for this statement type.

The parameter is ignored.

IMPLICIT or EXPLICIT required

The parameter *token* contains an incorrect value for the transaction type.

The parameter is ignored.

Expected statement

The statement type *token* was expected on this line but another statement was encountered.

The configuration file is declared incorrect and, after parsing the remaining lines, the Listener ends.

Unexpected statement

The statement type *token* encountered on this line was unexpected. This type of statement is not valid at this position in the file.

EZY1021I

The statement is ignored.

Maximum count exceeded

The current statement is a TRANSACTION statement and this statement has caused the number of these statements to exceed the limit specified by the MAXTRANS parameter of the LISTENER statement. The value that was specified is displayed by message EZY1021I, which is issued after this message.

The statement is ignored.

Value too long

The parameter value specified in *token* has a length that exceeds the maximum expected length for this parameter. The maximum expected length is displayed by message EZY1021I, which is issued after this message.

The parameter is ignored.

Invalid character

The parameter value specified in *token* contains a character that is not valid for this parameter such as an alphabetic character where a numeric value is required.

The parameter is ignored.

Value outside valid range

The parameter value in *token* specifies a numeric value which falls outside the allowed range of values. The allowed maximum and minimum values for this parameter are specified by messages EZY1021I and EZY1022I respectively, which are issued after this message.

The parameter is ignored.

Prior parameter conflict

The parameter value specified in *token* conflicts with a previous specification of this parameter on this statement.

If the parameter is mandatory then the statement is ignored. Otherwise, all definitions of this parameter are ignored.

Prior conflicting parameters

The parameter *token* has already been specified more than once on this statement, and these specifications were in conflict with each other.

If the parameter is mandatory, then the statement is ignored. Otherwise, all definitions of this parameter are ignored.

Redeclaration of parameter

The parameter *token* has already been specified on this statement, but the values are the same.

The statement is valid. This message can be avoided by specifying each parameter only once on a statement.

System Action: The entire configuration file is parsed and errors reported. If there are sufficient valid statements to specify all the configuration options, then the Listener continues, otherwise it ends.

User or Operator Response: Edit the configuration file and respecify the statements in error.

System Programmer Response: Assist the user as required.

Source Data Set: EZAIMSPS

Procedure Name: *varies*

EZY1021I *hh:mm:ss.th* **Maximum:** *max*

Explanation: This message is issued after message EZY1020E to identify the maximum allowed value for the parameter specified in that message.

System Action: See message EZY1020E for more information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: PRSCFG, ProcessParam

EZY1022I *hh:mm:ss.th* **Minimum:** *min*

Explanation: This message is issued after message EZY1020E to identify the minimum allowed value for the parameter specified in that message.

System Action: See message EZY1021E for more information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: ProcessParam

EZY1023E *hh:mm:ss.th* **File** *errtype* **error on** *ddname*, **RC=***rc*

Explanation: An I/O error occurred while accessing a file with the DDNAME *ddname*. The I/O activity is given by *errtype* and is one of the following:

OPEN
CLOSE
GET
PUT

Additional information about the error can be obtained from *rc* which has the following values and associated meanings:

- | | |
|---|--|
| 1 | Internal error. |
| | Return value for an end of file. Used internally and is not an error. If this value is displayed with this message then a program logic error has occurred. |
| 2 | File ABEND exit routine invoked. |
| 3 | File SYNAD exit routine invoked. |
| 4 | Internal error. |
| | This return code indicates that a GET or PUT has been performed on a file that has not been opened successfully. If this value is displayed with this message then a program logic error has occurred. |
| 5 | The record format for the file is not one of the supported types: FIXED or VARIABLE. |
| 6 | Internal error. |
| | A file structure pointer is not referencing a valid file structure. If this value is displayed with this message then a program error has occurred. |
| 7 | Error occurred while closing the file. |
| 8 | The <i>ddname</i> is not defined. |

System Action: The system action depends on *errtype* as follows:

Error	Description
-------	-------------

OPEN	The Listener opens the configuration file for input and the message log for output. If either file cannot be opened, the Listener ends.
-------------	---

Procedure Name: main, PRSCFG

CLOSE Errors associated with closing a file are ignored and processing continues.

Procedure Name: main, PRSCFG

GET If a GET error occurs before the Listener has been able to retrieve sufficient valid configuration statements to define all the configuration options, the Listener ends. Otherwise processing continues.

Procedure Name: PRSCFG

User or Operator Response: Take appropriate action based on *errtype* such as specifying the correct *ddnames* or specifying files with the correct attributes.

System Programmer Response: Assist the user as required.

EZY1024I • EZY1025W

Source Data Set: EZAIMSPS

Procedure Name: See the explanation above.

EZY1024I *event at hh:mm:ss.th on yyddd*

Explanation: One of the following events occurred at the date and time specified.

Listener invoked

The Listener has started, and configuration file processing is about to commence.

Procedure Name: main

Listener ended

The Listener has ended. Termination might have been initiated because of an operator request or because of errors. If an error condition caused the termination, the error will be described in a preceding message.

Procedure Name: main

No available sockets

The Listener currently has connections open on all the sockets obtained at startup as specified by the MAXACTSKT parameter of the LISTENER statement. The connection that has been requested by a client will only be accepted when a socket becomes available.

Procedure Name: ListenProcess

Security exit rejection

The user-supplied security exit has rejected a request. Further information about the client that initiated the request is supplied by messages EZY1031W, EZY1032I, and EZY1033I, which are issued after this message.

Procedure Name: ReadTRMPProcess

System Action: The system continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: See the explanation above.

EZY1025W *hh:mm:ss.th description: Code=rc*

Explanation: A system error has occurred. The nature of the error is given by *description*. The error is one of the following:

No storage available

A GETMAIN call, which is used to obtain storage for the TRANSACTION statements, has returned an error.

Refer to the GETMAIN macro description for an explanation of the return code *rc*.

Procedure Name: main, ListenProcess, PRSCFG

Memory deallocation unsuccessful

A FREEMAIN call has returned an error as evidenced by the return code.

Refer to the FREEMAIN macro description for an explanation of the return code *rc*.

Procedure Name: ReleaseProcess

CIB free unsuccessful

The START command CIB was not freed.

No action is required.

Procedure Name: main

Ignored unknown command

The listener received an unknown command. The command is ignored and processing continues.

Procedure Name: main

System Action: GETMAIN errors cause the Listener to end while FREEMAIN and CIB free errors are ignored as they might only occur during Listener shutdown.

User or Operator Response: Refer to the GETMAIN or FREEMAIN documentation and take action as indicated, such as increasing the region size.

System Programmer Response: Assist the user as required.

Source Data Set: EZAIMSPS

Procedure Name: See the explanation above.

EZY1026E *hh:mm:ss.th socket_error, socket__call* **ERRNO=errno**

Explanation: An error occurred during a socket call. The Listener processing at the time of the error and the socket call are given by *socket_error* and *socket_call*.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

A description of each of the errors is given below:

Unable to accept new connection, ACCEPT

An error occurred when the Listener attempted to complete a connection in response to a connection request from a client.

Procedure Name: ListenProcess

Unsuccessful binding to communication port, BIND

The Listener was unable to bind the port that it is to use for accepting connection requests from clients. The port is specified by the PORT parameter of the LISTENER statement.

Procedure Name: main

Socket CLOSE error:

An error occurred when the Listener attempted to close a connection.

Procedure Name: ListenProcess, ReleaseProcess

Error transferring connection to server, GIVESOCKET

An error occurred when the Listener attempted to pass control of the client connection to the server.

Procedure Name: CreServPro

Failed to connect to TCPIP address space, INITAPI

An error occurred while attempting to connect to the TCPIP address space. The TCPIP address space that is used is specified by the ADDRSPC parameter on the TCPIP statement.

Procedure Name: main

Failed setting socket to listen mode, LISTEN

The LISTEN call, used to enable the socket connected to the listening port for incoming connections, resulted in an error.

Procedure Name: main

Socket READ error:

An error occurred when the Listener attempted to receive data sent from a client program. Further information about the client is provided by message EZY1032I, which is issued after this message.

Procedure Name: ReadInputProcess, ReadTRMProcess

Error waiting for communications event, SELECT

The SELECT call, used to wait for connection requests and data for implicit transactions, has resulted in an error.

Procedure Name: main

Failed to obtain a socket to listen on, SOCKET

The Listener failed to acquire a socket for binding to the listening port.

Procedure Name: main

EZY1027W • EZY1028W

Server failed to take connection, TAKESOCKET

The Listener issued a GIVESOCKET and the server failed to issue a corresponding TAKESOCKET.

Note: ERRNO is not applicable to this error.

Socket WRITE error:

An error occurred when the Listener attempted to send data to a client program.

Procedure Name: WriteRSMProcess

System Action: The system action depends on the error. If the error occurs during startup processing or during a SELECT call, the Listener ends. Otherwise, the connection that received the error is closed.

User or Operator Response: For errors encountered during startup processing, check that the configuration file TCPIP and LISTENER parameters are correct. Otherwise, proceed based on the value of *errno*.

System Programmer Response: Assist the user as required.

Source Data Set: EZAIMSPS

Procedure Name: See the explanation above.

EZY1027W *hh:mm:ss.th* Unexpected condition on socket connection

Explanation: An unexpected exception condition or socket closure was detected on a socket connection while receiving data. Further information about the client is provided by message EZY1032I, which is issued after this message.

System Action: The connection is closed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: ReadInputProcess, SelectResponse, ReadTRMProcess

EZY1028W *hh:mm:ss.th description*

Explanation: A condition as described by *description* was detected by the Listener. The conditions are:

Invalid request received

A connection was accepted from a client, but the data received could not be interpreted as a transaction request message (TRM) segment.

The Listener sends a request status message (RSM) segment to the client and then closes the connection.

Procedure Name: ReadTRMProcess

Unknown transaction requested

The Transaction Request Message segment from a client specified a transaction that has not been defined to the Listener. Transactions are defined in the configuration file using TRANSACTION statements.

The Listener sends a request status message (RSM) segment to the client and then closes the connection.

Procedure Name: ReadTRMProcess

Buffer limit reached

The amount of data sent by a client for an implicit transaction has exceeded 32KB (including segment lengths and reserved bytes).

The Listener sends a request status message (RSM) segment to the client and then closes the connection.

Procedure Name: ReadInputProcess

IMS™ rollback performed

The Listener received a nonsuccessful status code when issuing an IMS call to build the IMS message for the requested transaction. The message has been removed by issuing a rollback call. This message will follow another message that has identified the cause of the error.

The Listener sends a request status message (RSM) segment, to the client, if possible, and then closes the connection. Sending the RSM is possible only if the error is detected before the GIVESOCKET call has been issued to pass the socket to the server.

Procedure Name: CreateServerProcess

numsktreq sockets requested, *numsktalloc* sockets allocated

The configuration file specified that *numsktreq* sockets be allocated to process connections; however, only *numsktalloc* sockets were made available by the socket interface. The number of sockets requested is specified by the MAXACTSKT parameter of the LISTENER statement.

Procedure Name: main

System Action: The system action is as indicated with each of the above entries.

User or Operator Response: Refer to the associated messages, EZY1032I and EZY1033I, which are issued in conjunction with this message to identify the client.

System Programmer Response: Assist the user as required.

Source Data Set: EZAIMSPS

Procedure Name: See the explanation above.

EZY1029E *hh:mm:ss.th error_description, IMS_function STATUS='status'*

Explanation: The nonsuccessful status code *status* was returned in response to an IMS/DC call. A description of the Listener processing at the time of the error and the IMS call function are given by *error_description* and *IMS_function* and are described below. This message appears with messages EZY1032I and EZY1033I, which provide further information about the client.

Error setting message destination, CHNG

The Listener attempted to set the destination of an alternate PCB to the transaction code specified in the transaction request message (TRM) segment in preparation for starting the transaction.

Procedure Name: CreateServerProcess

Error placing data onto transaction input queue, ISRT

The error occurred while inserting either the transaction initiation message (TIM) segment or a data segment if it was an implicit transaction.

Procedure Name: CreateServerProcess

Error committing transaction message, SYNC

The Listener had successfully inserted the message segments for the requested transaction but encountered the error when issuing the SYNC call.

Procedure Name: CreateServerProcess

Error rolling back transaction message, ROLB

The Listener encountered the error on a rollback call it issued to remove a message that it was creating when it encountered an error as described by the above two reasons.

Procedure Name: CreateServerProcess

System Action: Except for an error caused by a SYNC call, a request status message (RSM) segment indicating an IMS error is sent to the client and the connection is closed. For an error on the SYNC call, the Listener closes the connection. Sending the RSM is possible only if the error is detected before the GIVESOCKET call has been issued to pass the socket to the server. This call is issued just prior to the SYNC call.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: See the explanation above.

EZY1030E • EZY1034I

EZY1030E *hh:mm:ss.th* **Program Error #xx**

Explanation: An internal program error has occurred.

System Action: The Listener ends.

User or Operator Response: Record the number xx and contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: SelectResp

EZY1031W *hh:mm:ss.th* **Security exit RC=rc, Reason=rsn**

Explanation: The user-supplied security exit has rejected a request. This message appears with messages EZY1032I and EZY1033I, which provide further information about the client.

System Action: The Listener sends a request status message (RSM) segment to the client and closes the connection.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: ReadTRMProcess

EZY1032I *hh:mm:ss.th* **Peer: ipaddress**

Explanation: This message appears with other messages to further identify the client by providing the client's IP address.

System Action: See other messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: Various.

EZY1033I *hh:mm:ss.th* **Transaction: tran_code**

Explanation: This message appears with other messages to further identify the client by printing the transaction code *tran_code* that was requested by the client.

System Action: See other messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAIMSPS

Procedure Name: Various.

EZY1034I *hh:mm:ss.th* **AIB error: Retrn=xxxx Reasn=yyyy**

Explanation: The Listener detected an error when using the AIB interface to check the status of a requested transaction.

System Action: The Listener sends a request status message (RSM) segment to the client and closes the connection.

User or Operator Response: Refer to the section on AIB INQY call in *IMS/ESA Application Programming: DL/I Calls* for an explanation of the return and reason codes. See messages EZY1032I and EZY1033I for further information about the client.

System Programmer Response: Assist the user as required.

Source Data Set: EZAIMSPS

Procedure Name: CreateServerProcess

EZY1035I *hh:mm:ss.th tran_code* is unavailable or unknown

Explanation: The transaction *tran_code* that has been requested by a client is currently unavailable or is unknown to IMS. See messages EZY1032I and EZY1033I for further information about the client.

System Action: The Listener sends a request status message (RSM) segment to the client and closes the connection.

User or Operator Response: Make sure that a client program requests a transaction that has been defined to IMS.

System Programmer Response: Assist the user as required.

Source Data Set: EZAIMSPS

Procedure Name: CreateServerProcess

EZY1120E No TYPE specified

Explanation: The EZASMI macro requires a TYPE parameter. None was specified.

System Action: The system ends macro generation.

User or Operator Response: Specify a TYPE parameter on the EZASMI macro.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .TYP0

EZY1121E TYPE parameter is invalid

Explanation: The TYPE parameter specified for this macro is not valid.

System Action: The system ends macro generation.

User or Operator Response: Specify a valid TYPE for the macro and reissue the macro. See the *z/OS Communications Server: IP Programmer's Guide and Reference* for information about the macro and values for the TYPE parameter.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .TYP42

EZY1122E PPPP parameter required for type TTTT

Explanation: The parameter PPPP is required for the EZASMI macro when type TTTT is specified.

System Action: The system ends macro generation.

User or Operator Response: Refer to *z/OS Communications Server: IP Programmer's Guide and Reference* to determine the valid parameters for the macro type you specified.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .PG01L1, .PG03L1, .PG04L1, .PG05L1, .PG06L1, .PG07L1, .PG08L1, .PG08OKC, .PG08OKD, .PG08END, .PG09L1, .PG10L1, .PG11L1, .PG11OKD, .PG13L1, .PG13OKB, .PG14L1, .PG15L1, .PG15OKB, .PG18L1, .PG19L1, .PG19OKD, .PG20L1, .PG21A2, .PG21A3

EZY1123W PPPP parameter invalid for type TTTT

Explanation: The parameter PPPP is not required for the EZASMI macro when type TTTT is specified. It is ignored.

System Action: Macro generation continues.

User or Operator Response: Refer to *z/OS Communications Server: IP Programmer's Guide and Reference* to determine the valid parameters for the macro type you specified.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .TYP50, .PG01END, .PG02END, .PG03END, .PG04N1, .PG04END, .PG05END, .PG06END, .PG07END, .PG08OKA, .PG08OKB, .PG09L0, .PG09END, .PG10END, .PG11OKA, .PG11OKB, .PG11OKC, .PG11END, .PG12END, .PG13OKA, .PG13END, .PG14END, .PG15OKA, .PG15END, .PG16END, .PG17OKA, .PG17OKB, .PG17OKC, .PG17OKD, .PG17OKE, .PG17OKF, .PG17END, .PG18END, .PG19OKA, .PG19OKB, .PG19END, .PG20END, .PG21A1

EZY1124E xRETMSK Required if xSNDMSK specified

Explanation: On the SELECT, if a send mask is specified for read, write, or exception, the return mask must be specified for the same function.

System Action: The system ends macro generation.

User or Operator Response: Specify the return mask for the read, write, or exception function for which a send mask was specified.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .SELF4, .SELG4, .SELH4

EZY1125E Invalid AF specified, INET assumed

Explanation: A literal specified for AF on the SOCKET function is not valid. The macro assumed AF=INET.

System Action: The system ends macro generation.

User or Operator Response: The literal for AF on the SOCKET function must be either INET or IUCV.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .SOCA01

EZY1126E ECB not permitted for INITAPI

Explanation: The ECB parameter is not permitted for TYPE=INITAPI.

System Action: The system ends macro generation.

User or Operator Response: Remove the ECB parameter and reissue the macro.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .INI0

EZY1127E Invalid literal specified for PPPP

Explanation: The literal specified for parameter PPPP is not a valid value.

System Action: The system ends macro generation.

User or Operator Response: Refer to *z/OS Communications Server: IP Programmer's Guide and Reference* for the valid parameter values.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .FCNC01, .GSON05, .IOCC0E, .RCFF1, .SNDF1, .SDBT01, .SSON05

EZY1128E Invalid socket type specified, STREAM assumed.

Explanation: The socket type specified for a TYPE=SOCKET macro is not valid. SOCTYPE=STREAM is assumed.

System Action: The system ends macro generation.

User or Operator Response: Valid values for the SOCTYPE parameter are STREAM, DATAGRAM, and RAW.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .SOCCT3

EZY1129E No ECB specified for SYNC

Explanation: The TYPE=SYNC requires an ECB parameter.

System Action: The system ends macro generation.

User or Operator Response: Specify an ECB parameter on the TYPE=SYNC request and reissue the macro.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .ECB1

EZY1130E Invalid value for label

Explanation: The label specified is not a valid assembler label.

System Action: The system ends macro generation.

User or Operator Response: Specify a label that is a maximum of eight characters in length.

System Programmer Response: None.

Source Data Set: EZASMI

Procedure Name: .N0003

EZY1218E *mm/dd/yy hh:mm:ss* **PROGRAM** *programname* **DISABLED TRANID=** *transactionid* **PARTNER INET**
ADDR=*inetaddress* **PORT=***portnumber*

Explanation: The Listener checked the status of the program associated with the transaction. It was not enabled.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

programname is the name of the program that is associated with the transaction requested by the connecting client.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: Listener continues.

User or Operator Response: Use CEMT to determine and correct the status of the program.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

| **EZY1219E** *mm/dd/yy hh:mm:ss* **UNEXPECTED** *eventtype* **EVENT IN LISTENER** *transactionid* **FROM CLIENT IP**
 | **ADDRESS** *ipaddress* **PORT** *portnumber*

| **Explanation:** The CICS Listener was notified about an unexpected event.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| *eventtype* is the type of event: READ, WRITE, or EXCEPTION.

| *transactionid* is the name of the Listener's CICS transaction.

| *ipaddress* is the remote IP address of the client.

| *portnumber* is the remote port number of the client.

| **System Action:** The Listener closes the connection and continues processing.

| **User or Operator Response:** Contact the system programmer.

| **System Programmer Response:** If the event type is EXCEPTION, investigate whether or not the client is attempting to send out-of-band data. If necessary, have the client avoid sending out-of-band data. If the event type is not EXCEPTION or the client is not attempting to send out-of-band data, then contact the IBM Software Support Center.

| **Source Data Set:** EZACIC02

| **Procedure Name:** LISTENER

EZY1220E *mm/dd/yy hh:mm:ss* **READ FAILURE ON CONFIGURATION FILE** *PHASE=phase* **EIBRESP2=***response*

| **Explanation:** EZACIC21 was unable to read the IP CICS Sockets configuration file, EZACONFG.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

phase is the IP CICS Sockets initialization phase.

| *response* is the response from CICS when reading the IP CICS Sockets configuration file.

System Action: If the ABEND code is AEXY, then the listener ends normally. Otherwise, the listener ends with an ABEND code of EZAL.

User or Operator Response: Notify the CICS Systems Programmer.

System Programmer Response: Use the EIBRESP2 value to determine the problem and correct the file. See the *CICS User's Handbook* for information about EIBRESP2 values. If the EIBRESP2 value is zero, then the EZACONFG file has been defined as remote. If this is the configuration file you want, then verify that no CICS Sockets programs can run directly in the file owning region. This can cause the file to become disabled. Ensure that EZACIC20 is not in the file owning region PLT, and that the EZAC and EZAO transactions are unable to run directly in the file owning region. Attempts to open the file will fail if the file is defined with a value of YES specified in the ADD, DELETE, or UPDATE parameters in the CICS file definition in more than one CICS region.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1221E *mm/dd/yy hh:mm:ss* **CICS SOCKETS ENABLE FAILURE** **EIBRCODE** **BYTE2 =** *resp_code*

Explanation: The attempt to enable the task related user exit (TRUE) failed.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| *resp_code* is the CICS response code from attempting to enable IP CICS Sockets Task Related User Exit (TRUE).

System Action: Terminate the transaction.

User or Operator Response: Notify the CICS Systems Programmer.

System Programmer Response: Use the EIBRESP2 value to determine the problem and correct the file. An EIBRCODE BYTE2 value of 20 indicates the TRUE is already enabled. This will occur if you disable the interface using EZAO,STOP,CICS transaction and then immediately issue EZAO,START,CICS transaction before the Task Related User Exit (TRUE) is completely disabled from the previous EZAO,STOP,CICS transaction. See the *CICS User's Handbook* for information about EIBRCODEs.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1222E *mm/dd/yy hh:mm:ss* CICS/sockets registration failure return code= *return_code*

Explanation: The attempt to register the CICS Sockets Feature to z/OS failed.

System Action: Terminate the transaction.

User or Operator Response: Contact your System Administrator.

System Programmer Response: See the *z/OS MVS Programming: Product Registration* for information about the values for *return_code*.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1223E *mm/dd/yy hh:mm:ss* CICS/sockets attach failure return code = *return_code* REASON CODE = *reason_code*

Explanation: An attempt to attach one of the pool subtasks failed.

System Action: Stop attaching pool subtasks. The size of the pool is determined by the number of subtasks successfully attached.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: See the *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for information about the values for *return_code* and *reason_code* and make appropriate adjustments to your CICS environment.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1224I *mm/dd/yy hh:mm:ss* CICS/sockets initialization successful using *tasking_method*

Explanation: The CICS Sockets Interface has completed initialization successfully.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

tasking_method is the tasking method used to support the EZASOKET calls. The possible methods are:

Reusable MVS subtasks

Signifies that the IP CICS Sockets interface is using MVS subtasks from the pool generated according to the value specified on the NTASKS configuration parameter.

Non-reusable MVS subtasks

Signifies that the IP CICS Sockets interface is attaching an MVS subtask for each IP CICS Sockets-enabled application because NTASKS=0.

Open Transaction Environment

Signifies that the IP CICS Sockets interface is enabled to use CICS Open Transaction Environment. All EZASOKET calls will be processed on an Open API, L8, TCB. Programs calling EZASOKET should be coded to threadsafe programming standards and defined to CICS as CONCURRENCY(THREADSAFE) to benefit from this environment.

System Action: Continue with execution.

EZY1225E • EZY1228E

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1225E *mm/dd/yy hh:mm:ss* **STARTBR FAILURE ON CICS/_SOCKETS CONFIGURATION FILE PHASE=xx**
EIBRESP2=rrrrrr

Explanation: The STARTBR command used for the configuration file has failed.

System Action: Terminate the transaction.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Use the EIBRESP2 value to determine the problem. Check the CICS definition of the Configuration file to ensure the browse operation is permitted. See the *CICS User's Handbook* for information about EIBRESP2 values.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1226E *mm/dd/yy hh:mm:ss* **READNEXT FAILURE ON CICS/_SOCKETS CONFIGURATION FILE PHASE=xx**
EIBRESP2=rrrrrr

Explanation: The READNEXT command used for the configuration file has failed.

System Action: Terminate the transaction.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Use the EIBRESP2 value to determine the problem. Check the CICS definition of the Configuration file to ensure the browse operation is permitted. See the *CICS User's Handbook* for information about EIBRESP2 values.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1227E *mm/dd/yy hh:mm:ss* **CICS/_SOCKETS INVALID LISTENER TRANID = tran**

Explanation: The Listener transaction *tran* was not defined to CICS.

System Action: Terminate Listener Initialization.

User or Operator Response: Use CICS facilities to define the Listener transaction and program. Then use EZAO to start the Listener.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1228E *mm/dd/yy hh:mm:ss* **CICS/_SOCKETS LISTENER TRANSACTION *tran* DISABLED**

Explanation: The Listener transaction *tran* could not be started because it was disabled.

System Action: Terminate Listener Initialization.

User or Operator Response: Use CICS facilities to enable the transaction and then start the Listener using EZAO.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1229E *mm/dd/yy hh:mm:ss* **CICS SOCKETS LISTENER TRANSACTION *tran* NOT AUTHORIZED**

Explanation: The Listener transaction *tran* could not be started because it was not authorized.

System Action: Terminate Listener Initialization.

User or Operator Response: Use CICS facilities to authorize starting the Listener transaction and then start the Listener using EZAO.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1246E *mm/dd/yy hh:mm:ss* **CICS SOCKETS LISTENER PROGRAM ID *mmmmmmmmmm* INVALID**

Explanation: The Listener transaction could not be started because program *mmmmmmmmmm* is not defined.

System Action: Terminate Listener Initialization.

User or Operator Response: If the program ID is correct, use CICS facilities to define it. If it is not correct, use the EZAC transaction to correct the CICS Sockets Configuration file.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1247E *mm/dd/yy hh:mm:ss* **CICS SOCKETS LISTENER PROGRAM ID *mmmmmmmmmm* DISABLED**

Explanation: The Listener transaction could not be started because program *mmmmmmmmmm* is disabled.

System Action: Terminate Listener Initialization.

User or Operator Response: Use CICS facilities to enable the program and then use EZAO to start the Listener.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1250E *mm/dd/yy hh:mm:ss* **CICS/SOCKETS LISTENER *tran* NOT ON CONFIGURATION FILE**

Explanation: The Listener transaction *tran* is not defined on the CICS Sockets configuration file.

System Action: Terminate Listener Initialization.

User or Operator Response: If the Listener transaction name is correct, use the EZAC transaction to define it on the CICS Configuration file. If the name is not correct, correct it on the EZAO transaction.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1251E *mm/dd/yy hh:mm:ss* **CICS SOCKETS MODULE *mmmmmmmmmm* ABEND *xxxx***

Explanation: The CICS Sockets module *mmmmmmmmmm* has abended.

System Action: Terminate the transaction.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1252E *mm/dd/yy hh:mm:ss* **UNABLE TO LOAD EZASOH03 ERROR CODE=** *error_code* **REASON CODE=** *reason_code*

Explanation: During CICS Sockets initialization, the attempt to load module EZASOH03 failed.

System Action: Terminate Initialization.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: See the *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU* for information about the values for *error_code* and *reason_code* to determine why the module would not load. Also, look for associated MVS messages.

Source Data Set: EZACIC21

EZY1253E *mm/dd/yy hh:mm:ss* **CICS/SOCKETS LISTENER** *tran* **NOT ON CONFIGURATION FILE**

Explanation: An EZAO STOP LISTENER transaction was run with an invalid Listener name.

System Action: Present the panel to correct the name.

User or Operator Response: Correct the name and retry termination.

System Programmer Response: None.

Source Data Set: EZACIC22

Procedure Name: TERMINATION

EZY1254E *mm/dd/yy hh:mm:ss* **CACHE FILE ERROR RESP2 VALUE ***** CALL # ***

Explanation: An error occurred on a cache file operation.

System Action: Return to the calling program with an error response.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Use the RESP2 value to determine the error and correct the cache file. See the *CICS User's Handbook* for information about RESP2 values.

Source Data Set: EZACIC25

Procedure Name: DOMAIN NAME SERVER FUNCTION

EZY1255E *mm/dd/yy hh:mm:ss* **TEMPORARY STORAGE ERROR RESP2 VALUE ***** CALL # ***

Explanation: An error occurred on a temporary storage operation in EZACIC25.

System Action: Return to the calling program with an error response.

User or Operator Response: Use the RESP2 value to determine the error. Contact the IBM Software Support Center. See the *CICS User's Handbook* for information about RESP2 values.

System Programmer Response: None.

Source Data Set: EZACIC25

Procedure Name: DOMAIN NAME SERVER FUNCTION

EZY1256E *mm/dd/yy hh:mm:ss* **CICS SOCKETS INTERFACE NOT ENABLED PRIOR TO LISTENER STARTUP**

Explanation: An attempt to start a Listener was made when the CICS Sockets Interface was inactive.

System Action: Return error and terminate transaction EZAO.

User or Operator Response: Use transaction EZAO to start the CICS Sockets Interface prior to starting the Listener.

System Programmer Response: None.

Source Data Set: EZACIC21

Procedure Name: INITIALIZATION

EZY1258I *module* **ENTRY POINT IS** *address*

Explanation: This message displays the entry point address of a module.

module is the name of the module.

address is the entry point address of the module.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC01, EZACIC02

EZY1259E *mm/dd/yy hh:mm:ss* **IOCTL CALL FAILURE TRANSACTION=***transactionid* **TASKID=***tasknumber*
ERRNO=*errno*

Explanation: Listener transaction *transactionid* experienced a failure on the IOCTL call.

In the message text:

mm/dd/yy

The date (month/day/year) of the message.

hh:mm:ss

The time (hours:minutes:seconds) of the message.

transactionid

The name of the transaction under which the Listener is executing.

tasknumber

The CICS task number of the Listener task.

errno The UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: If the error is during initialization of the Listener, then the Listener transaction *transactionid* terminates. Otherwise, the Listener closes the socket that was being processed and resumes normal processing.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1260E *mm/dd/yy hh:mm:ss* **EZACIC03 ATTACH FAILED GPR15=***xxxxxxx* **ERRNO=***errno* **TRAN=***tran*
TASK=*cicstask*

Explanation: An ATTACH for an MVS subtask has failed. The reason code is in GPR 15.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The task related user exit (TRUE) for this transaction is disabled. The transaction abends with an AEY9.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Determine the cause for the ATTACH failure and correct.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

| **EZY1261I** *mm/dd/yy hh:mm:ss* **EZACIC03 ATTACH SUCCESSFUL, TCB ADDRESS=** *tcbaddr* **TERM=***term*
TRAN=*tran* **TASK=***cicstask*

Explanation: An ATTACH for an MVS subtask was successful. This message is produced only for Listeners and for those tasks which cannot be accommodated within the pool of reusable tasks.

- | *mm/dd/yy* is the date (month/day/year) of the message.
- | *hh:mm:ss* is the time (hours:minutes:seconds) of the message.
- | *tcbaddr* is the address of the Task Control Block (TCB) being attached.
- | *term* is the CICS terminal ID associated with the CICS transaction identified by *tran*.
- | *tran* is the name of the CICS transaction that was requested.
- | *cicstask* is the task number of the CICS transaction identified by *tran*.

System Action: Processing continues.

- | **User or Operator Response:** If this message happens frequently, increase the size of the reusable task pool, NTASKS, for this CICS. Increasing NTASKS appropriately will prevent overhead incurred with attaching the subtask. See the EZACICD TYPE parameter in *z/OS Communications Server: IP CICS Sockets Guide* for information the NTASKS value.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1262E *mm/dd/yy hh:mm:ss* **GWA ADDRESS INVALID UEPGAA=***xxxxxxxx* **TRAN=***tran* **TASK=***cicstask*

Explanation: The task related user exit (TRUE) detected an invalid GWA address.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Use EZAO to stop (immediate) and start the CICS Sockets Interface. If the problem repeats, contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1263E *mm/dd/yy hh:mm:ss* **TIE ADDRESS INVALID UEPGAA=***xxxxxxxx* **TRAN=***tran* **TASK=***cicstask*

Explanation: The task related user exit (TRUE) detected an invalid TIE address.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Use EZAO to stop (immediate) and start the CICS Sockets Interface. If the problem repeats, contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1264E *mm/dd/yy hh:mm:ss* **FLAG WORD ADDRESS INVALID UEPFLAGS=** *xxxxxxxx* **ERRNO=***errno*
TRAN=*tran* **TASK=***cicstask*

Explanation: The task related user exit (TRUE) detected an invalid flag word address.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Use EZAO to stop (immediate) and start the CICS Sockets Interface. If the problem repeats, contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1265E *mm/dd/yy hh:mm:ss* **CICS VERSION UNSUPPORTED GWACIVRM=xxxx ERRNO=errno TRAN=tran**
TASK=cicstask

Explanation: The task related user exit (TRUE) detected a version of CICS which it does not support. The CICS version must be 3 or above.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: The CICS Sockets Interface requires CICS V3R3 or later.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1267E *mm/dd/yy hh:mm:ss* **ROUTING TASK FUNCTION INVALID UERTIFD=xx ERRNO=errno TRAN=tran**
TASK=cicstask

Explanation: The task related user exit (TRUE) detected an invalid routing task function.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: If this happens repeatedly, use EZAO to STOP (immediate) the CICS Sockets Interface and then START it. If it still happens, contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1268E *mm/dd/yy hh:mm:ss* **SAVE AREA ADDRESS INVALID UEPHSM= xxxxxxxx ERRNO=errno**
TRAN=tran TASK=cicstask

Explanation: The task related user exit (TRUE) detected an invalid save area address.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1269E *mm/dd/yy hh:mm:ss* **PARM LIST ADDRESS INVALID GPR1= xxxxxxxx ERRNO=errno TRAN=tran**
TASK=cicstask

Explanation: The task related user exit (TRUE) detected an invalid parameter list on a call request from the CICS application program.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

EZY1270E • EZY1272E

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Check the application program calls to the CICS Sockets Interface to ensure that each call has the correct number and type of parameters.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1270E *mm/dd/yy hh:mm:ss* PARM nn ADDRESS INVALID ADDRESS= xxxxxxxx ERRNO=errno TRAN=tran
TASK=cicstask

Explanation: The task related user exit (TRUE) detected an invalid parameter address on a call request from the CICS application program. nn is the number of the parameter.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Check the application program calls to the CICS Sockets Interface to ensure that the parameter addresses are valid (not zero). This problem is most common in assembler language and C applications.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1271E *mm/dd/yy hh:mm:ss* TOKERR=xxxxxxx ERRNO=errno TRAN=tran TASK=cicstask

Explanation: The task related user exit (TRUE) detected a token error on an internal token used to coordinate CICS transaction activity with TCP/IP activity.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1272E *mm/dd/yy hh:mm:ss* INVALID SOCKET/FUNCTION CALL FUNCTION= xxxx ERRNO=errno
TRAN=tran TASK=cicstask

Explanation: A call to EZASOKET specified in invalid function.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Correct the call and retry.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: task related user exit (TRUE)

EZY1273E *mm/dd/yy hh:mm:ss* **IUCV SOCK/FUNC TABLE INVALID FUNCTION=** xxxx **ERRNO=***errno*
TRAN=*tran* **TASK=***cicstask*

Explanation: A call to EZACICAL specified a function that was not valid.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Correct the call and retry.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1274E *mm/dd/yy hh:mm:ss* **INCORRECT EZASOKET PARM COUNT FUNCTION=** xxxx **ERRNO=***errno*
TRAN=*tran* **TASK=***cicstask*

Explanation: A call to EZASOKET specified in invalid number of parameters.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Correct the call and retry.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1275E *mm/dd/yy hh:mm:ss* **MONITOR CALLS NOT SUPPORTED UERTFID=xx** **ERRNO=***errno* **TRAN=***tran*
TASK=*cicstask*

Explanation: The task related user exit (TRUE) detected a monitor call which is not supported for this version of CICS.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1276E *mm/dd/yy hh:mm:ss* **EDF CALLS NOT SUPPORTED UERTFID=xx** **ERRNO=***errno* **TRAN=***tran*
TASK=*cicstask*

Explanation: The task related user exit (TRUE) detected an EDF (Execute Diagnostic Facility) call. This TRUE does not support EDF calls.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE is disabled and the task abends with an AEY9.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

EZY1277I • EZY1282E

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1277I *mm/dd/yy hh:mm:ss* **EZACIC03 DETACHED TCB ADDRESS=xxxxxxx ERRNO=errno TRAN=tran**
TASK=cicstask

Explanation: An attached subtask is terminating.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The TRUE detaches the MVS subtask.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1278I *mm/dd/yy hh:mm:ss* **EZACIC03 DETACH SUCCESSFUL TCB ADDRESS= xxxxxxxx TRAN=tran**
TASK=cicstask

Explanation: An attached subtask is terminating.

System Action: The TRUE detaches the MVS subtask.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1279E *mm/dd/yy hh:mm:ss* **INVALID SYNC PT COMMAND DISP=xx TRAN=tran TASK=cicstask**

Explanation: The task related user exit (TRUE) Detected an invalid Sync Point command.

System Action: Disable the TRUE and return to the caller.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1280E *mm/dd/yy hh:mm:ss* **INVALID RESYNC COMMAND DISP=xx TRAN=tran TASK=cicstask**

Explanation: The task related user exit (TRUE) Detected an invalid Resync command.

System Action: Disable the TRUE and return to the caller.

User or Operator Response: Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC01

EZY1282E *mm/dd/yy hh:mm:ss* **10999 ABEND reasonxx**

Explanation: The ESTAE processing in EZACIC03 could not be completed because of *reasonxx*.

System Action: Allow the ABEND to percolate.

User or Operator Response: Contact the IBM Software Support Center. See the *CICS User's Handbook* for information about abend codes.

System Programmer Response: None.

Source Data Set: EZACIC03

Procedure Name: MVS SUBTASK

EZY1285E *mm/dd/yy hh:mm:ss* **CICS/sockets listener transaction NOT ON CONFIGURATION FILE**

Explanation: The Listener attempting to start does not have a description record on the CICS Sockets configuration file.

System Action: Listener terminates.

User or Operator Response: Contact CICS Systems Programmer.

System Programmer Response: Add the Listener to the configuration file using EZAC and retry.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1286E *mm/dd/yy hh:mm:ss* **READ FAILURE ON CICS/sockets configuration file transaction EIBRESP= rrrrr**

Explanation: The Listener could not read the configuration file.

System Action: Listener terminates.

User or Operator Response: Contact CICS Systems Programmer.

System Programmer Response: Use the CICS APR to interpret the value of EIBRESP. If the file is not known to CICS, perform the installation steps for the configuration file.

See the *CICS User's Handbook* for information about EIBRESP values.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1287E *mm/dd/yy hh:mm:ss* **EZYCIC02 GETMAIN FAILURE FOR VARIABLE STORAGE transaction EIBRESP=rrrrr**

Explanation: EZACIC02 could not obtain the variable storage it requires to execute.

System Action: Listener terminates.

User or Operator Response: Contact CICS Systems Programmer.

System Programmer Response: Use the CICS APR to interpret the value of EIBRESP. Correct your CICS configuration as indicated.

See the *CICS User's Handbook* for information about EIBRESP values.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1288E *mm/dd/yy hh:mm:ss* **CICS sockets module mmmmmmmmm ABEND aaaa**

Explanation: An abend has occurred in module *mmmmmmmmmm* of the CICS Sockets Interface.

System Action: Listener terminates.

User or Operator Response: See the *CICS User's Handbook* for information about abend codes. Contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1289E *mm/dd/yy hh:mm:ss* **CICS LISTENER TRANSACTION *tran* TERMINATING**

Explanation: The Listener is terminating. This could be a normal shutdown situation or a failure related to the Listener socket. If it is the latter, a previous message will describe the failure.

System Action: Continue termination of the Listener.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1290I *mm/dd/yy hh:mm:ss* **LISTENER TRANSACTION *tran* STARTING**

Explanation: Transaction *tran*, Listener program EZACIC02 has been given control.

System Action: Listener *tran* continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1291I *mm/dd/yy hh:mm:ss* **LISTENER TRANSACTION *tran* TASKID= *cicstask* ACCEPTING REQUESTS VIA PORT *pppppp***

Explanation: Listener transaction *tran* is now available to receive connection requests on port *pppppp*.

System Action: Listener *tran* continues

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1292E *mm/dd/yy hh:mm:ss* **CANNOT START LISTENER, TRUE NOT ACTIVE TRANSACTION= *tran* TASKID= *cicstask* EIBRCODE BYTE3=rr**

Explanation: The initialization of the CICS Sockets Interface did not complete successfully and this Listener cannot continue.

System Action: Listener transaction *tran* terminates.

User or Operator Response: If EZAO is being used to start the Listener, ensure that the CICS Sockets interface has successfully completed initialization first. If this happens during automatic initialization, look for other messages which would indicate why the initialization of the CICS Sockets Interface failed.

See the *CICS User's Handbook* for information about EIBRCODEs.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1293E *mm/dd/yy hh:mm:ss* **INITAPI CALL FAILURE TRANSACTION=*tran* TASKID= *cicstask* ERRNO=*errno***

Explanation: Listener transaction *tran* experienced a failure on the INITAPI call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Programmer Response: None.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1294E *mm/dd/yy hh:mm:ss* **SOCKET CALL FAILURE TRANSACTION=** *tran* **TASKID=** *cicstask* **ERRNO=** *errno*

Explanation: Listener transaction *tran* experienced a failure on the SOCKET call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Programmer Response: None.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1295E *mm/dd/yy hh:mm:ss* **BIND CALL FAILURE TRANSACTION=** *tran* **TASKID=** *cicstask* **ERRNO=** *errno*

Explanation: Listener transaction *tran* experienced a failure on the BIND call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

Notes:

1. An ERRNO=13 could indicate that the port and jobname specified in the PORT statement in *hlq*.TCPIP.PROFILE does not match the port and jobname used by the CICS Listener.
2. An ERRNO=48 could indicate that the port is not reserved in *hlq*.TCPIP.PROFILE.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1296E *mm/dd/yy hh:mm:ss* **LISTEN CALL FAILURE TRANSACTION=** *tran* **TASKID=** *cicstask* **ERRNO=** *errno*

Explanation: Listener transaction *tran* experienced a failure on the LISTEN call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1297E *mm/dd/yy hh:mm:ss* **GETCLIENTID CALL FAILURE TRANSACTION=***tran* **TASKID=** *cicstask*
ERRNO=*errno*

Explanation: Listener transaction *tran* experienced a failure on the GETCLIENTID call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1298E *mm/dd/yy hh:mm:ss* **CLOSE FAILURE TRANID=** *tran* **TASKID=** *cicstask* **ERRNO=** *errno*

Explanation: Listener transaction *tran* experienced a failure on the CLOSE call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: Listener transaction *tran* continues.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1299E *mm/dd/yy hh:mm:ss* **SELECT CALL FAILURE TRANSACTION=** *tran* **TASKID=** *xxxxx* **ERRNO=** *errno*

Explanation: Listener transaction *tran* experienced a failure on the SELECT call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1300E *mm/dd/yy hh:mm:ss* **RECV FAILURE TRANSID=** *transactionid* **TASKID=** *tasknumber* **ERRNO=** *errno*
INET ADDR=*inetaddress* **PORT=***portnumber*

Explanation: The Listener transaction *transactionid* experienced a failure on the RECV call.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the Listener transaction performing the RECV Socket.

tasknumber is the CICS task number assigned to the CICS transaction *transactionid*.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction *transactionid* continues.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1301E *mm/dd/yy hh:mm:ss* **CONNECTION CLOSED BY CLIENT TRANSACTION=** *transactionid* **PARTNER**
INET ADDR= *ipaddr* **PORT=** *port*

Explanation: A remote client connected to the CICS Listener but then closed the connection before sending the entire amount of data required by the Listener as determined by the MINMSGL standard Listener configuration parameter or the MSGLEN enhanced Listener configuration parameter.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the transaction name of the CICS Listener.

ipaddr is the internet address of the remote client.

port is the port number of the remote client.

System Action: The Listener transaction *transactionid* continues.

User or Operator Response: Correct the client program.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1302I *mm/dd/yy hh:mm:ss* **READ TIMEOUT PARTNER INET ADDR=***inetaddress* **PORT=***portnumber*

Explanation: The initial message from the client did not arrive within the read timeout value specified for this Listener in the CICS Sockets configuration file.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener closes the connection socket and does not attempt to start a server transaction.

User or Operator Response: Determine the cause of the delay and correct it.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1303I *mm/dd/yy hh:mm:ss* **EZACIC02 GIVESOCKET TIMEOUT TRANS** *transactionid* **PARTNER INET**
ADDR=*inetaddress* **PORT=***portnumber*

Explanation: The started server transaction did not perform the takesocket within the timeout value specified for this Listener in the CICS Sockets configuration file.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

EZY1304I • EZY1306E

System Action: Send an error message to the client and close the socket.

User or Operator Response: Determine the reason for the delay in the server transaction. Possible causes are an overloaded CICS system or excessive processing in the server transaction before the takesocket is issued. Correct the situation and retry.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1304I *mm/dd/yy hh:mm:ss* **UNEXPECTED INPUT EVENT TRANSACTION** *transactionid* **PARTNER INET**
ADDR=inetaddress **PORT=portnumber**

Explanation: The Listener received data from the client after the end of the transaction input message.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener ignores this data.

User or Operator Response: Ensure that the minimum message length specification for this Listener in the CICS Sockets Configuration file is correct. If it is, determine why the client is sending this additional data.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1305E *mm/dd/yy hh:mm:ss* **UNEXPECTED EXCEPTION EVENT TRANS** *transactionid* **PARTNER INET**
ADDR=inetaddress **PORT=portnumber**

Explanation: The Listener received an exception event on this connection other than the event showing a successful takesocket was issued by the server.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: Ignore the event.

User or Operator Response: Ensure the client is not doing anything that would cause an exception event such the use of out-of-band data.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1306E *mm/dd/yy hh:mm:ss* **SECURITY EXIT** *mmmmmmmm* **IS NOT DEFINED** **TRANID= tran**
TASKID=xxxxxxx

Explanation: The security exit specified for this Listener in the CICS Sockets configuration file is not defined to CICS.

System Action: Close the socket and terminate the connection.

User or Operator Response: Use CICS RDO to define the security exit.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1307E *mm/dd/yy hh:mm:ss* **MAXIMUM # OF SOCKETS USED TRANS=** *tran* **TASKID=** *cicstask* **ERRNO=** *errno*

Explanation: All of the sockets allocated to Listener transaction *xxxx* are in use.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: The ACCEPT call is delayed until a socket is available.

User or Operator Response: Use the EZAC transaction to increase the number of sockets allocated Listener *tran* and then stop and restart Listener transaction *tran*.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1308E *mm/dd/yy hh:mm:ss* **ACCEPT CALL FAILURE TRANSACTION=** *tran* **TASKID=** *cicstask* **ERRNO=** *errno*

Explanation: Listener transaction *tran* experienced a failure on the ACCEPT call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

System Action: Listener transaction *tran* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1309E *mm/dd/yy hh:mm:ss* **GIVESOCKET FAILURE TRANS** *transactionid* **TASKID=** *tasknumber* **ERRNO=** *errno*
INET ADDR= *inetaddress* **PORT=** *portnumber*

Explanation: The Listener transaction *transactionid* experienced a failure on the GIVESOCKET call.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

tasknumber is the CICS task number assigned to the CICS transaction *transactionid*.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction *transactionid* terminates.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1310E *mm/dd/yy hh:mm:ss* **IC VALUE NOT NUMERIC TRANID=*transactionid* PARTNER INET**
ADDR=*inetaddress* PORT=*portnumber*

Explanation: The interval specified in the transaction input message contains one or more non-numeric characters.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The interval is ignored, and the transaction is started immediately.

User or Operator Response: Correct the client program which is sending this transaction input message.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1311E *mm/dd/yy hh:mm:ss* **CICS TRANID *transactionid* NOT AUTHORIZED PARTNER INET**
ADDR=*inetaddress* PORT=*portnumber*

Explanation: The transaction name specified in the transaction input message is not RSL authorized.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The transaction is not started.

User or Operator Response: Correct the CICS transaction definition if the transaction should be authorized or the client program if it is sending the wrong transaction name.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1312E *mm/dd/yy hh:mm:ss* **SECURITY EXIT *mmmmmmmmmm* CANNOT BE LOADED TRANID= *tran***
TASKID=*cicstask*

Explanation: Listener transaction *tran* experienced a failure when it attempted to load security exit program *mmmmmmmmmm*.

System Action: Listener transaction *tran* continues but the server transaction associated with this transaction input message is not started.

User or Operator Response: Use CEMT to determine the status of the exit program and correct whatever problems are found.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1313E *mm/dd/yy hh:mm:ss* **LISTENER NOT AUTHORIZED TO ACCESS SECURITY EXIT** *mmmmmmmmmm*
TRANID= *tran* **TASKID=***xxxxxxxx*

Explanation: Listener transaction *tran* is not authorized to access security exit program *mmmmmmmmmm*.

System Action: Listener transaction *tran* continues but the server transaction associated with this transaction input message is not started.

User or Operator Response: If the security exit program name is incorrect, use EZAC to correct the definition of this Listener on the CICS Sockets Configuration file. If the security exit program is correct, use the CICS RDO facility to authorize Listener transaction *xxxx* to use security exit program *mmmmmmmmmm*.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1314E *mm/dd/yy hh:mm:ss* **SECURITY EXIT** *mmmmmmmmmm* **IS DISABLED** **TRANID=** *tran* **TASKID=***xxxxxxxx*

Explanation: Security exit program *mmmmmmmmmm* is disabled.

System Action: Listener transaction *tran* continues but the server transaction associated with this transaction input message is not started.

User or Operator Response: Use CEMT to enable the security exit program.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1315E *mm/dd/yy hh:mm:ss* **INVALID TRANSID** *transactionid* **PARTNER INET ADDR=***inetaddress*
PORT=*portnumber*

Explanation: The transaction input message from the client specified transaction *transactionid* but this transaction is not defined to CICS.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

| *transactionid* is the name of the transaction that was requested by the connecting client. The *transactionid* field will be
 | blank if no printable name was passed by the client or the security exit.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues but the server transaction associated with this transaction input message is not started.

User or Operator Response: If the transaction name is incorrect, correct the client program. If the transaction name is correct, correct the CICS transaction definition.

| **System Programmer Response:** If *transactionid* is blank, then there is a possible mismatch because the Listener is
 | expecting the first message segment to start with a transaction name but it does not. A packet trace might be helpful
 | in determining whether there is such a mismatch. For example, if the packet trace shows that the first message
 | segment starts with X'160300' or X'160301' then possibly a **clienthello** message was received, which indicates that
 | there is an Application Transparent Transport Layer Security (AT-TLS) policy on the client side of the TCP connection
 | but no matching AT-TLS policy (or AT-TLS is not enabled) on the Listener side of the TCP connection. This would
 | need to be addressed by the AT-TLS administrator. See Application Transparent Transport Layer Security (AT-TLS)
 | Data Protection in *z/OS Communications Server: IP Configuration Guide* and Diagnosing AT-TLS problems in *z/OS*
 | *Communications Server: IP Diagnosis Guide* for more information.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1316E *mm/dd/yy hh:mm:ss* **TRANSID** *transactionid* **IS DISABLED PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: Transaction *transactionid* is disabled.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues but the server transaction associated with this transaction input message is not started.

User or Operator Response: Use CEMT to enable the server transaction.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1317E *mm/dd/yy hh:mm:ss* **TRANSID** *transactionid* **IS NOT AUTHORIZED PARTNER INET**
ADDR=inetaddress PORT=portnumber

Explanation: The Listener transaction *transactionid* is not authorized to start the transaction name specified in the transaction input message.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The transaction is not started.

User or Operator Response: Authorize Listener transaction *transactionid* to start the transaction.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1318E *mm/dd/yy hh:mm:ss* **TD START SUCCESSFUL QUEUEID= qqqq**

Explanation: The Listener transaction started a server transaction through transient data queue qqqq.

System Action: Listener transaction continues and the server transaction is ready to start.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1319E *mm/dd/yy hh:mm:ss* **QIDERR FOR TD DESTINATION** *queueName* **PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction through transient data queue *queueName*. DFHRESP was QIDERR.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

queuename is the name of the transient data queue that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: If the queue name is incorrect, correct the client program sending this transaction input message. If the queue name is correct, correct the CICS Destination Control Table.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1320E *mm/dd/yy hh:mm:ss* **I/O ERROR FOR TD DESTINATION** *queuename* **PARTNER INET**
ADDR=inetaddress PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction through transient data queue *queuename*. DFHRESP was IOERR.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

queuename is the name of the transient data queue that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1321E *mm/dd/yy hh:mm:ss* **LENGTH ERROR FOR TD DESTINATION** *queuename* **PARTNER INET**
ADDR=inetaddress PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction through transient data queue *queuename*. DFHRESP was LENGERR.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

queuename is the name of the transient data queue that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer. The minimum length for this queue should be greater than 72.

System Programmer Response: Change definition of Transient Data Queue to accommodate length of this message.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1322E *mm/dd/yy hh:mm:ss* **TD DESTINATION** *queueName* **DISABLED PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction through transient data queue *queueName*. DFHRESP was DISABLED.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

queueName is the name of the transient data queue that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Use CEMT to enable the destination.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1323E *mm/dd/yy hh:mm:ss* **TD DESTINATION** *queueName* **OUT OF SPACE PARTNER INET**
ADDR=inetaddress PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction through transient data queue *queueName*. DFHRESP was NOSPACE.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

queueName is the name of the transient data queue that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Allocate space for this Transient Data Queue.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1324E *mm/dd/yy hh:mm:ss* **TD START FAILED QUEUE ID=queueName** **PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction through transient data queue *queueName*.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

queueName is the name of the transient data queue that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Determine the problem with the Transient Data Queue and correct it.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1325I *mm/dd/yy hh:mm:ss* **START SUCCESSFUL TRANID=***transactionid* **PARTNER INET ADDR=***inetaddress*
PORT=*portnumber*

Explanation: The Listener transaction was able to start a CICS transaction *transactionid* transient data queue.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1326E *mm/dd/yy hh:mm:ss* **START I/O ERROR TRANID=***transactionid* **PARTNER INET ADDR=***inetaddress*
PORT=*portnumber*

Explanation: The Listener transaction was unable to start a CICS transaction *transactionid*. DFHRESP was IOERR.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Determine the cause of the I/O error and correct it.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1327E *mm/dd/yy hh:mm:ss* **START TRANSACTION ID** *transactionid* **INVALID PARTNER INET**
ADDR=*inetaddress* **PORT=***portnumber*

Explanation: The Listener transaction was unable to start a CICS transaction *transactionid*. DFHRESP was TRANSIDERR.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

EZY1328E • EZY1330E

System Programmer Response: Check the transaction definition in RDO to ensure it is correct.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1328E *mm/dd/yy hh:mm:ss* **START TRANSACTION ID** *transactionid* **NOT AUTHORIZED PARTNER INET**
ADDR=inetaddress **PORT=portnumber**

Explanation: The Listener transaction was unable to start a CICS transaction *transactionid*. DFHRESP was NOTAUTH.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: If the transaction ID is incorrect, correct the client program which sent this transaction input message. If the transaction ID is correct, authorize Listener transaction to start this transaction.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1329E *mm/dd/yy hh:mm:ss* **START FAILED (99) TRANSID=transactionid** **PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: The Listener transaction was unable to start a CICS transaction *transactionid*. DFHRESP was 99.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Check the transaction definition in RDO. Look for associated messages in the MSGUSR queue, which might indicate why the transaction would not start.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1330E *mm/dd/yy hh:mm:ss* **IC START SUCCESSFUL TRANID=transactionid** **PARTNER INET**
ADDR=inetaddress **PORT=portnumber**

Explanation: The Listener transaction was able to start a CICS transaction *transactionid*.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1331E *mm/dd/yy hh:mm:ss* **IC START I/O ERROR** *TRANID=transactionid* **PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: Listener transaction was unable to start a CICS transaction *transactionid*. DFHRESP was IOERR.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Look for other messages in the MSGUSR queue, which provide specific information on the I/O error and correct the problem.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1332E *mm/dd/yy hh:mm:ss* **IC START INVALID REQUEST** *TRANID=transactionid* **PARTNER INET**
ADDR=inetaddress PORT=portnumber

Explanation: Listener transaction was unable to start a CICS transaction *transactionid*. DFHRESP was INVREQ.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: Listener transaction continues.

User or Operator Response: Collect the messages written to the console and MSGUSR queue, client input data, and a SOCKAPI component trace and contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1333E *mm/dd/yy hh:mm:ss* **IC START FAILED** *TRANID=transactionid* **PARTNER INET ADDR=inetaddress**
PORT=portnumber

Explanation: Listener transaction was unable to start a CICS transaction *transactionid*.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

EZY1334E • EZY1336E

portnumber is the connecting client's port number.

System Action: Listener transaction continues.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Check the RDO definition of the transaction. Collect the messages written to the console and MSGUSR queue, client input data, and a SOCKAPI component trace and contact the IBM Software Support Center.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1334E *mm/dd/yy hh:mm:ss* **INVALID USER TRANID=***transactionid* **PARTNER INET ADDR =** *inetaddress*
PORT = *portnumber*

Explanation: This message indicates that the user security exit has given the Listener an invalid USERID field.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The server transaction does not start.

User or Operator Response: Correct the invalid USERID in the security exit.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1335E *mm/dd/yy hh:mm:ss* **WRITE FAILED ERRNO=***errno* **TRANID=***transactionid*. **PARTNER INET**
ADDR=*inetaddress* **PORT=***portnumber*

Explanation: Listener transaction had a failure on a WRITE command.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1336E *mm/dd/yy hh:mm:ss* **TAKESOCKET FAILURE TRANS** *transactionid* **TASKID=***tasknumber* **ERRNO=***errno*
INET ADDR=*inetaddress* **PORT=***portnumber*

Explanation: The Listener transaction had a failure on a TAKESOCKET command.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: The Listener transaction continues.

User or Operator Response: Use the *errno* value to determine the cause of the failure.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1337E *mm/dd/yy hh:mm:ss* CICS IN QUIESCE, LISTENER TERMINATING TRANSID= *tran* TASKID= *cicstask*

Explanation: Listener transaction *tran* is terminating because it detected a CICS quiesce in progress.

System Action: Listener transaction *tran* terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1338E *mm/dd/yy hh:mm:ss* PROGRAM *programname* NOT FOUND TRANID=*transactionid* PARTNER INET ADDR=*inetaddress* PORT=*portnumber*

Explanation: The Listener checked the status of the program associated with the transaction. It was not found.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

programname is the name of the program which is associated with the transaction requested by the connecting client.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client.

portnumber is the connecting client's port number.

System Action: Listener continues.

User or Operator Response: If *transactionid* is incorrect, correct the client program that sent the transaction input message. If the transaction ID is correct, check the transaction and program definitions in CICS.

System Programmer Response: None.

Source Data Set: EZACIC02

EZY1339E *mm/dd/yy hh:mm:ss* EXIT PROGRAM (EZACIC01) IS NOT ENABLED. DISABLE IGNORED TERM=*term* TRAN=*tranxxx*

Explanation: A termination of the CICS Sockets Interface was requested but the interface is not enabled.

System Action: The termination request is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC22

EZY1340E • EZY1344I

Procedure Name: TERMINATION

EZY1340E *mm/dd/yy hh:mm:ss* API ALREADY QUIESCING DUE TO PREVIOUS REQ. EZAO IGNORED
TERM=*term* TRAN=*tranxxx*

Explanation: A request for a quiesce of the CICS Sockets interface has been made but one is already in progress.

System Action: Ignore the second request.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC22

Procedure Name: TERMINATION

EZY1341E *mm/dd/yy hh:mm:ss* API ALREADY IN IMMED MODE DUE TO PREV. REQ. EZAO IGNORED
TERM=*term* TRAN=*tranxxx*

Explanation: A request for an immediate of the CICS Sockets interface has been made but one is already in progress.

System Action: Ignore the second request.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC22

Procedure Name: TERMINATION

EZY1342I *mm/dd/yy hh:mm:ss* DISABLE DELAYED UNTIL ALL USING TASKS COMPLETE TERM=*term*
TRAN=*tranxxx*

Explanation: A quiesce is in progress and is waiting for the completion of all outstanding CICS tasks using the CICS sockets interface.

System Action: Continue with the quiesce.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC22

Procedure Name: TERMINATION

EZY1343I *mm/dd/yy hh:mm:ss* CICS/SOCKETS INTERFACE IMMEDIATELY DISABLED TERM=*term*
TRAN=*tranxxx*

Explanation: A request for immediate termination of the CICS Sockets Interface has been successfully completed.

System Action: Terminate the CICS Sockets Interface.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC22

Procedure Name: TERMINATION

EZY1344I *mm/dd/yy hh:mm:ss* CICS/SOCKETS INTERFACE QUIESCENTLY DISABLED TERM=*term*
TRAN=*tranxxx*

Explanation: A request for deferred termination of the CICS Sockets Interface has been successfully completed.

System Action: Terminate the CICS Sockets Interface.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACIC22

EZY1345E *mm/dd/yy hh:mm:ss* **CICS/sockets WLM REGISTER FAILURE. RETURN CODE = *return_code*,
GROUP = *groupname*, LISTNER = *list***

Explanation: The CICS Listener received an error response when attempting to register WLM group with the Workload manager.

mm/dd/yy hh:mm:ss

Date and time of the message.

return_code

The return code from the WLM registration.

groupname

Name of the WLM group.

list

Name of the CICS Listener.

System Action: The Listener continues initialization but will not use *groupname* to participate in workload connection balancing.

User or Operator Response: Verify that the WLM group name is correct and correctly defined to the Workload manager. If it is incorrect, either change it in the EZACICD TYPE=LISTENER macro that was used to define the Listener, or change it via the EZAC transaction. See the *z/OS MVS Programming: Workload Management Services* for more information about *return_code*.

System Programmer Response: None

Source Data Set: EZACIC12

EZY1346E *mm/dd/yy hh:mm:ss* **CICS sockets WLM DEREGISTER FAILED RETURN CODE = *return_code*,
GROUP = *groupname*, LISTNER = *list***

Explanation: The CICS Listener received an error response when attempting to deregister WLM group with the Workload manager.

mm/dd/yy hh:mm:ss

Date and time of the message.

return_code

The return code from the WLM deregistration.

groupname

Name of the WLM group.

list

Name of the CICS Listener.

System Action: The Listener continues termination.

User or Operator Response: See the *z/OS MVS Programming: Workload Management Services* for more information about *return_code*.

System Programmer Response: None.

Source Data Set: EZACIC12

EZY1347I *mm/dd/yy hh:mm:ss* **PROGRAM *programname* ASSUMED TO BE AUTOINSTALLED
TRANID=*transactionid* IP ADDR=*inetaddress* PORT=*portnumber***

Explanation: The Listener checked the status of the program associated with the transaction. It was not found. Since program autoinstall is active in the CICS region, the Listener assumes that the program definition will automatically be installed by CICS.

mm/dd/yy

The date (month/day/year) of the message.

EZY1348E • EZY1349E

hh:mm:ss

The time (hours:minutes:seconds) of the message.

programname

The name of the undefined program which is associated with the transaction requested by the connecting client.

transactionid

The name of the transaction that was requested by the connecting client.

inetaddress

The internet address of the connecting client.

portnumber

The connecting client's port number.

System Action: Listener continues.

User or Operator Response: None.

System Programmer Response: Verify that the program name in the transaction definition is correct. Verify that the program is intended to be autoinstalled rather than explicitly defined in the PPT.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1348E *mm/dd/yy hh:mm:ss* **INVALID SOCKET FUNCTION** *function* **ERRNO** *errno* **TRAN** *tranid* **TASK** *taskid*

Explanation: The task related user exit (TRUE) detected an invalid socket function on a call request from the CICS application program.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

function is the invalid socket function.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

tranid is the name of the CICS transaction.

taskid is the CICS task ID number.

System Action: The TRUE is disabled and the task abends with an AEY9 CICS abend code.

User or Operator Response: Correct the invalid socket function and retry.

The most probable *errno* is 10011 "INVALID SOCKET FUNCTION". If the socket function name appears correct, ensure that the application padded the function call with blanks.

System Programmer Response: None.

Source Data Set: EZACIC01

Procedure Name: Task Related User Exit (TRUE)

EZY1349E *mm/dd/yy hh:mm:ss* **UNABLE TO OPEN CONFIGURATION FILE TRANSACTION=***transactionid*
EIBRESP2=*eibresp2*

Explanation: The CICS Listener received an abnormal response from CICS when attempting to open the CICS Sockets configuration file (EZACONFG) using an EXEC CICS SET FILE call.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction under which the Listener is executing.

eibresp2 is the EIBRESP2 value returned by CICS on the EXEC CICS SET FILE call as described in *CICS System Programming Reference*.

System Action: The Listener ends.

User or Operator Response: Contact the CICS system programmer.

System Programmer Response: Use the *CICS System Programming Reference* to interpret the value of EIBRESP2. If the file is not known to CICS, perform the installation steps for the configuration file.

Source Data Set: EZACIC02

Procedure Name: LISTENER

EZY1350E *mm/dd/yy hh:mm:ss* NOT AUTHORIZED TO USE *api_function*, action IGNORED. TERM=*termid*
TRAN=*transid*

Explanation: The IP CICS Sockets interface uses a CICS EXTRACT EXIT command to determine whether the IP CICS Sockets Task Related User Exit (TRUE) is enabled. This action is performed by IP CICS Sockets interface initialization and shutdown programs, the Listener, and by any user application linking to the IP CICS domain name server module.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

api_function is the CICS command performed.

action is the action intended.

- ENABLE means the IP CICS Sockets interface is being enabled.
- DISABLE means the IP CICS Sockets interface is being disabled.
- STARTUP means the IP CICS Sockets interface is being started.

termid is the terminal ID where the transaction receiving the error is executing.

transid is the name of the transaction that is incurring the security violation.

System Action:

- If the TRUE is being enabled when the IP CICS Sockets Interface is initializing, then the enable action is ignored and the interface is not activated.
- If the TRUE is being disabled when the IP CICS Sockets Interface is shutting down, then the disable action is ignored and the interface remains active.
- If the IP CICS Sockets interface is being started, then the startup action is ignored and the interface remains inactive.

User or Operator Response: Contact the CICS system programmer.

System Programmer Response: Ensure that the user ID being used is allowed at least UPDATE access to the EXITPROGRAM resource.

Source Data Set: EZACIC02, EZACIC21, EZACIC22

Procedure Name: Listener, Initialization, Shutdown

EZY1351E *mm/dd/yy hh:mm:ss* EXIT PROGRAM (EZACIC01) IS NOT ENABLED, action IGNORED.
TERM=*termid* TRAN=*transid*

Explanation: The IP CICS Sockets interface uses a CICS ENABLE PROGRAM command to enable the IP CICS Sockets Task Related User Exit (TRUE). This action is performed by IP CICS Sockets interface initialization.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

action is the action intended.

- ENABLE means the IP CICS Sockets interface is being enabled.
- DISABLE means the IP CICS Sockets interface is being disabled.

termid is the terminal ID where the transaction receiving the error is executing.

EZY1352E • EZY1353E

transid is the name of the transaction that is incurring the security violation.

System Action: The IP CICS Sockets Interface is not initialized.

User or Operator Response: Contact the CICS system programmer.

System Programmer Response: Ensure that the user ID being used is allowed at least UPDATE access to the EXITPROGRAM resource.

Source Data Set: EZACIC21

Procedure Name: Initialization

EZY1352E *mm/dd/yy hh:mm:ss* **SUBTASK ENDED UNEXPECTEDLY TRANSACTION=** *transactionid* **TASKID=** *taskid*

Explanation: The current tasks CICS Sockets subtask ended unexpectedly. This is probably caused by an ABEND of the subtask.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the CICS transaction whose subtask ended unexpectedly.

taskid is the CICS task number of the task whose subtask ended unexpectedly.

System Action: The CICS Sockets interface is disabled for the current task. Any subsequent CICS Sockets calls by that task will result in CICS ABEND code AEY9. Other tasks are not affected.

User or Operator Response: Contact the CICS Systems Programmer.

System Programmer Response: Check the console log for previous messages that explain what happened to the subtask.

Source Data Set: EZACIC01

Procedure Name: TASK RELATED USER EXIT (TRUE)

EZY1353E *mm/dd/yy hh:mm:ss* **COMMA MISSING AFTER IC TRANS ID =** *transactionid* **PARTNER IP ADDR =** *inetaddress* **PORT =** *portnumber*

Explanation: The listener did not find a comma delimiter after the interval control (IC) start type indicator in the client's transaction request message.

mm/dd/yy is the date (month/day/year) of the message.

hh:mm:ss is the time (hours:minutes:seconds) of the message.

transactionid is the name of the transaction that was requested by the connecting client.

inetaddress is the internet address of the connecting client

portnumber is the connecting client's port number.

System Action: The listener will not start the transaction specified by the client's transaction request message and will end the connection. This message will also be returned to the client.

User or Operator Response: Ensure that a comma delimiter separates the IC start type and the IC start time. See Listener input format in *z/OS Communications Server: IP CICS Sockets Guide* for information about the client's transaction request message.

System Programmer Response: None.

Source Data Set: EZACIC02

Procedure Name: LISTENER

| **EZY1354I** *mm/dd/yy hh:mm:ss* **CICS/SOCKETS CICS TRACING IS status**

| **Explanation:** This message shows the status of changing IP CICS Sockets CICS tracing and is issued when one of the following occurs:

- | • The operator issued the EZAO,START,TRACE transaction.
- | • The operator issued the EZAO,STOP,TRACE transaction.
- | • The CICS Master User Trace Flag is specified as OFF and the IP CICS Sockets TRACE configuration is specified as YES.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| *status* is the status of CICS tracing for the IP CICS Sockets Interface.

- | • ENABLED indicates that the IP CICS Sockets interface will generate CICS trace data when CICS tracing is active.
- | • DISABLED indicates that the IP CICS Sockets interface will not generate CICS trace data.

| **System Action:** When *status* is ENABLED, IP CICS Sockets will generate CICS trace data when CICS tracing is active. When *status* is DISABLED, IP CICS Sockets will not generate CICS trace data.

| **User or Operator Response:** None.

| **System Programmer Response:** None.

| **Source Data Set:** EZACIC00, EZACIC01

| **Procedure Name:** TRC00000, SUB05100

| **EZY1355I** *mm/dd/yy hh:mm:ss* **CICS/SOCKETS TCBLIM EXCEEDS MAXOPENTCBS**

| **Explanation:** IP CICS Sockets has determined that the value specified for TCBLIM exceeds the value of MAXOPENTCBS allowed at the time the interface was enabled. TCBLIM will be forced to the same value as MAXOPENTCBS.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| **System Action:** IP CICS Sockets TCBLIM will default to the value of MAXOPENTCBS. IP CICS Sockets processing continues.

| **User or Operator Response:** Contact the CICS system programmer.

| **System Programmer Response:** Adjust the value specified by the TCBLIM configuration option using one or more of the following methods:

- | • Specify an appropriate TCBLIM value on the EZACICD TYPE=CICS,TCBLIM= macro.
- | • Specify an appropriate TCBLIM value using the EZAC Configuration transaction.
- | • Specify an appropriate TCBLIM value dynamically by using the EZAO Operator transaction.
- | • Specify an appropriate MAXOPENTCBS value using the CICS System Initialization parameters.
- | • Specify an appropriate MAXOPENTCBS value using the CICS Master Terminal transaction, CEMT SET DISPATCHER MAXOPENTCBS.

| Refer to the following sections in *z/OS Communications Server: IP CICS Sockets Guide*:

- | • Building the configuration data set with EZACICD for information about using the EZACICD macro.
- | • Configuration transaction (EZAC) for information about the EZAC Configuration transaction.
- | • SET function and INQUIRE function for information about the EZAO Operator transaction.
- | • the EZACICD TYPE parameter in *z/OS Communications Server: IP CICS Sockets Guide* for a description of the TCBLIM parameter.

| Refer to the *CICS System Definition Guide* for a description of the MAXOPENTCBS parameter. Refer to *CICS Supplied Transactions* for information about using the CEMT transaction.

EZY1356E • EZY1358E

| **Source Data Set:** EZACIC21
| **Procedure Name:** Initialization

| **EZY1356E** *mm/dd/yy hh:mm:ss* **CICS/sockets TCBLIM HAS BEEN REACHED**

| **Explanation:** The number of IP CICS Sockets-enabled CICS tasks using an Open API, L8, TCB is equal to the value specified by the TCBLIM configuration option.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| **System Action:** The IP CICS Sockets interface will suspend any new tasks until one of the following actions occur:

- The IP CICS Sockets TCBLIM value is increased.
- Existing transactions using IP CICS Sockets end.

| This message will be issued only when the interface detects that it has reached TCBLIM. EZY1360I will be issued when this condition is relieved.

| **User or Operator Response:** Contact the CICS system programmer.

| **System Programmer Response:** Use the CICS Master Terminal transaction, CEMT INQ TASK HVALUE(ATTTCBLIM), to determine which IP CICS Sockets-enabled CICS transactions are subject to TCBLIM. Either take action to reduce the IP CICS Sockets work load or increase the IP CICS Socket TCBLIM configuration option. You can use the EZAO,SET,CICS Operator transaction to dynamically increase TCBLIM. The new value you set for the TCBLIM configuration option must be less than or equal to the value specified by MAXOPENTCBS.

| **Source Data Set:** EZACIC01

| **Procedure Name:** SUB16000

| **EZY1357I** *mm/dd/yy hh:mm:ss* **TRANSIENT DATA QUEUE SPECIFIED ON ERROR TD IS NOT DEFINED TO CICS**

| **Explanation:** IP CICS Sockets has determined that the CICS transient data queue specified by the ERROR TD configuration option was not defined to the CICS region where the IP CICS Sockets interface is enabled.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| **System Action:** The CSMT transient data queue will be used for reporting all IP CICS Sockets interface messages. CSMT is the default CICS transient data queue name.

| **User or Operator Response:** Contact the CICS system programmer.

| **System Programmer Response:** Ensure that the CICS transient data queue specified by the ERROR TD configuration option is properly defined to CICS.

| See Transient data definition in *z/OS Communications Server: IP CICS Sockets Guide* for more information.

| **Source Data Set:** EZACIC21

| **Procedure Name:** Initialization

| **EZY1358E** **10999 ABEND - IP CICS SOCKETS USING OTE**

| **Explanation:** IP CICS Sockets has incorrectly called the MVS subtask wrapper module when the interface was enabled to use CICS Open Transaction Environment.

| **System Action:** The IP CICS Sockets interface will stop.

| **User or Operator Response:** Contact the CICS system programmer.

| **System Programmer Response:** Contact the IBM Software Support Center. See the *CICS User's Handbook* for information about abend codes.

| **Source Data Set:** EZACIC03

| **Procedure Name:** MVS SUBTASK

| **EZY1359I** *mm/dd/yy hh:mm:ss* **CICS/sockets APPLICATIONS WILL USE THE QR TCB**

| **Explanation:** IP CICS Sockets has determined that CICS FORCEQR=YES is specified.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| **System Action:** CICS will force all user application programs, including those enabled to IP CICS Sockets, that are specified as threadsafe to run under the CICS Quasi-Reentrant (QR) TCB, as if they were specified as quasi-reentrant programs.

| **User or Operator Response:** Contact the CICS system programmer.

| **System Programmer Response:** If you do not want to incur the overhead of CICS switching Open API-enabled tasks back to the QR TCB, then change the value of FORCEQR to NO. Refer to the *CICS System Definition Guide* for more information about the FORCEQR CICS System Initialization parameter. Refer to *CICS Supplied Transactions* for more information about the CICS Master Terminal transaction that is used to dynamically change the FORCEQR setting.

| **Source Data Set:** EZACIC21

| **Procedure Name:** Initialization

| **EZY1360I** *mm/dd/yy hh:mm:ss* **CICS/sockets TCBLIM CONDITION HAS BEEN RELIEVED**

| **Explanation:** IP CICS Sockets enable transactions are no longer suspended due to TCBLIM.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| **System Action:** Any new or suspended IP CICS Sockets work will now be processed without being suspended due to IP CICS Sockets being at TCBLIM.

| **User or Operator Response:** None.

| **System Programmer Response:** None.

| **Source Data Set:** EZACIC01

| **Procedure Name:** SUB16000, Task termination

| **EZY1361E** *mm/dd/yy hh:mm:ss* **CICS/TS OPEN TRANSACTION ENVIRONMENT SUPPORT IS NOT AVAILABLE**

| **Explanation:** The IP CICS Sockets OTE configuration parameter is specified as YES. IP CICS Sockets determined that the CICS environment that is required to support the exploitation of CICS Open Transaction Environment by IP CICS Sockets is not available.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| **System Action:** The IP CICS Sockets interface is not enabled to use CICS Open Transaction Environment.

| **User or Operator Response:** Contact the System Programmer.

| **System Programmer Response:** Perform one of the following:

- | • Upgrade the level of CICS to support Open Transaction Environment. The CICS Open Transaction Environment requires CICS/TS V2R2 or later.
- | • Change the IP CICS Sockets interface configuration to use MVS subtasks when configuring it by using the EZAC configuration transaction or the EZACICD macro.

| **Source Data Set:** EZACIC21

| **Procedure Name:** Initialization

| **EZY1362E** *mm/dd/yy hh:mm:ss* **CICS/sockets START OF LISTENER** *transactionid* **FAILED RESP1=** *resp1*
 | **RESP2=***resp2*

| **Explanation:** CICS Sockets attempted to start the specified listener, but the EXEC CICS START command failed with the RESP1 and RESP2 values listed in the message text.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| *transactionid* is the transaction name of the listener that the CICS Sockets attempted to start.

| *resp1* is the RESP1 value returned by the EXEC CICS START transaction.

| *resp2* is the RESP2 value returned by the EXEC CICS START transaction.

| **System Action:** The CICS Listener does not start.

| **User or Operator Response:** None.

| **System Programmer Response:** Refer to the description of the START command in the *CICS Application Programming Reference* for information about why the START command failed.

- | • If the RESP2 value is 8 or 9, then the problem is related to the USERID parameter in the definition of the listener. Verify that the USERID parameter is correct. See Setting up and configuring CICS TCP/IP in *z/OS Communications Server: IP CICS Sockets Guide* for a description of the USERID parameter.
- | • If the RESP2 value is 8, then the USERID parameter of the listener definition specifies a user ID that is not known to RACF®. Therefore, either change the USERID parameter or define the user ID to RACF.
- | • If the RESP2 value is 9, then the user ID under which the EXEC CICS START was issued does not have SURROGAT security access to the user ID that is specified in the USERID parameter. For example, if the failure occurs during CICS PLT processing, then the PLT user ID does not have SURROGAT security access to the listener's user ID. Refer to the *CICS RACF Security Guide* for more information.

| **Source Data Set:** EZACIC21

| **Procedure Name:** INITIALIZATION

| **EZY1364I** *mm/dd/yy hh:mm:ss* **LISTENER** *transactionid* **DETECTED THAT TTLS IS** *status* **ON STACK** *tcpname*

| **Explanation:** The CICS Listener is defined with a GETTID parameter of YES which indicates that the listener is requested to attempt to obtain the connecting client certificates and user IDs from Application Transparent Transport Layer Security (AT-TLS). If status is DISABLED, then AT-TLS is disabled in the TCP/IP stack. Therefore, the listener is unable to obtain client certificates and user IDs as requested by the GETTID parameter. If status is ENABLED, then AT-TLS has been enabled in the TCP/IP stack, making it possible for the listener to obtain client certificates and user IDs.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| *transactionid* is the name of the listeners CICS transaction.

| *status* is the status of AT-TLS in the TCP/IP stack. *status* is either DISABLED or ENABLED.

| *tcpname* is the name of the TCP/IP stack.

| **System Action:** The listener continues its normal processing, which includes attempting to obtain client certificates and User IDs.

| **User or Operator Response:** Contact the system programmer.

| **System Programmer Response:** No response is needed if status is ENABLED. If status is DISABLED, then verify that the GETTID parameter of YES is correct in the listener definition. If so, request that your AT-TLS administrator investigate why AT-TLS is not enabled in the TCP/IP stack. See Setting up and configuring CICS TCP/IP in *z/OS Communications Server: IP CICS Sockets Guide* for a description of the GETTID parameter.

| See Application Transparent Transport Layer Security (AT-TLS) Data Protection in *z/OS Communications Server: IP Configuration Guide* and Diagnosing AT-TLS problems in *z/OS Communications Server: IP Diagnosis Guide* for more information.

| **Source Data Set:** EZACIC02

| **Procedure Name:** LISTENER

| **EZY1366E** *mm/dd/yy hh:mm:ss* **CICS/_SOCKETS LISTENER TRANSACTION *tranid* IS ALREADY ACTIVE**

| **Explanation:** The IP CICS Sockets Listener determined that another listener with the same transaction ID is already active.

| *mm/dd/yy* is the date (month/day/year) of the message.

| *hh:mm:ss* is the time (hours:minutes:seconds) of the message.

| *tranid* is the CICS transaction identifier of the duplicate IP CICS Sockets Listener.

| **System Action:** The IP CICS Sockets Listener that issued this message ends.

| **User or Operator Response:** Contact the system programmer.

| **System Programmer Response:** Change the Listeners CICS transaction identifier or port number to ensure that the definition is unique. See Setting up and configuring CICS TCP/IP in *z/OS Communications Server: IP CICS Sockets Guide* for more information about configuring the IP CICS Sockets Listener.

| **Source Data Set:** EZACIC02

| **Procedure Name:** Initialization

EZY1372W **Dataset *.TCPIP.DATA not found**

Explanation: The system could not find a *.TCPIP.DATA data set to parse during initialization of the REXX socket interface.

System Action: The program continues and attempts to use the default information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: INITFWRN

EZY1373I **RXSOCKET - REXX/_SOCKETS (a component of 5694-A01):**

EZY1374I **REXX support for the TCP/IP Socket Interface**

EZY1381I **Allocating DD=*device* DSN=*data_set_name***

Explanation: The indicated device is allocating the specified data set for task-related storage.

System Action: REXX Sockets continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: main

EZY1382E ***task* failed; RC=*return_code*, INFO=*dynalloc_info*, ERROR=*dynalloc_error***

Explanation: The indicated task was unsuccessful. The **INFO=*dynalloc_info*, ERROR=*dynalloc_error*** portion of the message only appears when *task* is ALLOCATE to indicate dynamic allocation.

task is the name of the task that failed.

return_code portion of this message indicates why the task was unsuccessful.

dynalloc_info is the DYNALLOC information reason code. See the *z/OS MVS Programming: Authorized Assembler*

EZY1383I • EZY1386E

Services Guide for information about information reason codes from DYNALLOC.

dynalloc_error is the DYNALLOC error reason code. See the *z/OS MVS Programming: Authorized Assembler Services Guide* for information about error reason codes from DYNALLOC.

System Action: Rexx Sockets continues.

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 *z/OS Communications Server: IP and SNA Codes* to determine the cause of the error and respond as indicated. See the *z/OS MVS Programming: Authorized Assembler Services Guide* for information about information reason codes and error reason codes from DYNALLOC.

Source Data Set: SOCMVS

Procedure Name: main

EZY1383I Dataset not found

Explanation: A requested data set was not accessible to the host.

System Action: Rexx Sockets continues.

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

System Programmer Response: Make sure that the data set is in storage accessible to the host.

Source Data Set: SOCMVS

Procedure Name: main

EZY1384E Return code *rc* from IRXSUBCM function

Explanation: The specified function failed while trying to get storage.

System Action: Rexx Sockets halts.

User or Operator Response: Increase the region size for the application.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: main

EZY1385E Unable to acquire LOC=BELOW GETMAIN storage for QSAM DCB and exits

Explanation: The Rexx Sockets failed while trying to get storage.

System Action: Rexx Sockets halts. Application continues.

User or Operator Response: Increase the region size for the application.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: main

EZY1386E Return code *rc* from GETMAIN SVC

Explanation: The Rexx Sockets failed while trying to get storage.

System Action: Rexx Sockets halts. Application continues.

User or Operator Response: Increase the region size for the application.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: main

EZY1387E Product 5694-A01 (MVS TCP/IP) is not available

Explanation: The Rexx Sockets failed when trying to access the TCP/IP JES control block.

System Action: Rexx Sockets halts. Application continues.

User or Operator Response: Contact system programmer.

System Programmer Response: Make TCP/IP available to the application.

Source Data Set: SOCMVS

Procedure Name: main

EZY1388E Return code xx from IDENTIFY SVC

Explanation: The Rexx Sockets could not identify the entry point for TCPERROR.

System Action: Rexx Sockets halts. Application continues.

User or Operator Response: Contact system programmer.

System Programmer Response: Verify that Rexx Sockets has been installed correctly.

Source Data Set: SOCMVS

Procedure Name: main

EZY1389E Unable to acquire Dynamic Save Area storage

Explanation: The function GETMAIN, which is used to acquire storage, was unsuccessful. No storage is allocated.

System Action: Rexx Sockets halts.

User or Operator Response: Increase the region size for the application and retry.

System Programmer Response: Assist the user as necessary.

Source Data Set: SOCMVS

Procedure Name: main

EZY1391E Unable to initialize Global DSA

Explanation: The host was unable to initialize the global dynamic storage area (DSA). This can occur due to lack of storage accessibility to the host.

System Action: RXSOCKET halts. TCPIP continues.

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

System Programmer Response: Increase the size of the storage region available to the application.

Source Data Set: SOCMVS

Procedure Name: main

EZY1400I Unable to establish ESTAE exit; processing continues

Explanation: Rexx Sockets was unable to establish an Error State Exit.

System Action: Rexx sockets continues.

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

System Programmer Response: Verify that Rexx Sockets was installed correctly.

Source Data Set: SOCMVS

Procedure Name: main

EZY1401I • EZY1424I

EZY1401I **REXX/SOCKETS anchor located via IRXSUBCM** *address*

Explanation: This debug message displays the address of persistent storage used as the anchor.

System Action: REXX sockets continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: INSTEP4

EZY1410E **Abend** *abend code* **detected in REXX/Sockets at** *location*

EZY1411E **REXX/Sockets loaded at** *rexx sockets location*; **Failing PSW:** *psw*

EZY1412E *register numbers : register_values*

Explanation: REXX/Sockets detected an abend. The abend code and the offset within the routine that caused the abend are displayed, followed by the location of the REXX/Sockets program, the Program Status Word (PSW) at the time of the abend, and the value of the general purpose registers.

System Action: REXX/Sockets ends and returns control to the application.

User or Operator Response: Determine the cause of the abend by referring to *z/OS MVS System Codes* manual for an explanation of the System abend codes. For more information, refer to the messages that usually follow these messages and are generated by the TSO/REXX abend handler.

System Programmer Response: None.

Source Data Set: SOCMVS

Procedure Name: STAEGOOD

EZY1421I *service* **SERVICE CALL Socket call:** *socket_call*

EZY1422I **Return_value** *retval* **hex**

EZY1423I **Return_code** *retcode* **hex**

EZY1424I **Reason_code** *rsncode* **hex**

Explanation: These messages show the return information from the z/OS UNIX System Services used to implement the REXX sockets API when SOCKDEBUG is specified in the TCPIP DATA data set. The BPXxxxx service call is *service* and the name of that service is *socket_call*.

For example: EZY1421I BPx1SOC SERVICE CALL Socket call: socket

These services, and the associated return information, are documented in *z/OS UNIX System Services Programming: Assembler Callable Services Reference*.

System Action: REXX/Sockets continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RXSOCKET

Procedure Name: TRACER

EZY1870E **CANNOT LOAD** *module* **ROUTINE.**

Explanation: An attempt to load the indicated module into virtual storage was unsuccessful. Control is returned to the operating system with an error code indicating that a LOAD failure occurred.

System Action: The task initialization function is terminated.

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

System Programmer Response: Check that the identified module resides in a library that is accessible to the MVS platform code. See *z/OS Communications Server: IP Configuration Reference* for information about required library residence for TCPIP components.

Source Data Set: MVPMAIN

Procedure Name: Mainline code

EZY1876I *server* **STACK FUNCTIONS STARTED WITH PARAMETER** *profile*.

Explanation: Informational message displayed every time a TCPIP server or TCPIP is invoked. The *profile* parameters for each started server are displayed.

System Action: The system continues invoking TCPIP and required services.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: Mainline code

EZY1877I *module* **STACK FUNCTIONS SHUTDOWN IS COMPLETE, RC = rc.**

Explanation: This message displays on the operator console after shutdown has completed for the load indicated by *module*. If the function indicated by *module* terminated normally, then RC = 0. If the function indicated by *module* terminated abnormally, then RC = 200.

System Action: The function indicated by *module* terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPMAIN

Procedure Name: Mainline code

EZY1889I **MVP I/O Interrupt from** *device number* **CSW=channel status word.**

Explanation: This message displays if the debug option is used. The MVS platform processed an I/O interrupt on the indicated device. If the device number displayed is 01, the device is a fake device.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPIOINT

Procedure Name: ProcIO

EZY1890I **MVP CheckUserID for** *user id*, **rc = rc, rsn = reason.**

Explanation: This message displays if the debug option is used. The MVS platform called the SAF interface to check a user ID. This message displays the user ID that was entered and a return and reason code. If the return code is not zero, the user ID is not valid.

System Action: TCPIP continues.

EZY1891I • EZY1895I

User or Operator Response: Check and make sure that the syntax of the user ID is correct and that the ID is set up in the MVS system. See your system administrator for help with this message.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPUTIL

Procedure Name: CHECKUID

EZY1891I MVP CCW to start=*ccw*

Explanation: This message displays if the debug option is used. The MVS platform channel command word (CCW) to be issued is displayed. CCWs have either of two different formats. The two formats do not differ in the information contained in the CCW but only in the arrangement of the fields within the CCW. The formats are designated format 0 and format 1. Format 0 CCWs can be located anywhere in the first 16Mb of main storage. Format 1 CCWs can be located anywhere in main storage.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPEXCP

Procedure Name: SCC1

EZY1892I MVP HaltIO to device *device* issued.

Explanation: This message displays if the debug option is used. The MVS platform halt I/O routine (MVPEXCPH) was invoked for the indicated device.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPEXCP

Procedure Name: MVPEXCPH

EZY1894I MVP OCM Invoked Operator (or) Start (or) Stop (or) Modify (or) Unknown

Explanation: This message is issued if the debug option is used. **Operator** indicates that the operator communication routine is invoked. **Start**, **Stop**, and **Modify** specifies which command was requested. **Unknown** indicates that MVP does not recognize the command received.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: MVPOCM1

EZY1895I MVP DEBUG table entry: *keyword*= ON (or) OFF

Explanation: This message displays when the debug option is used. The indicated command and its switch setting are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: Dval

EZY1896I MVP *action* **Control Register 0 = value.**

Explanation: This message displays if the debug option is used. The read control register 0 and the set control register 0 routines are called to update control register 0. These are part of a list of tasks required to disable the reception of virtual machine communication facility (VMCF). For more information, refer to the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

action is the action the function performed against the value in Control Register 0. The action will be either Read or Set.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDOSM

Procedure Name: READC0, SETC0

EZY1897I MVP **Received Ext Interrupt Code=ext_code.**

Explanation: This message displays if the debug option is used. MVP received the indicated external interrupt code.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPIOINT

Procedure Name: ProcExt

EZY1898I MVP **Setting Ext Interrupt Type=type Value=value**

Explanation: This message is displayed if the debug option is used. The PC external interrupt routine was set. The interrupt type and the clock comparator value is displayed. Valid type is external timer at (X'1004').

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPEXTR

Procedure Name: MVPEXTR

EZY1899I MVP **Ext Interrupt subclass ext_code is disabled, C0=current control register 0.**

Explanation: This message is displayed if the debug option is used. An external interrupt code was disabled and the current control register zero (CR0) is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPIOINT

Procedure Name: ProcExt

EZY1900I MVP Clock Comparator Interrupt overridden by a new one.

Explanation: This message displays if the debug option is used. The external interrupt has been replaced by a new interrupt with a later clock comparator value.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPIOINT

Procedure Name: ProcExt

EZY1901I MVP Close failed, *number I/Os outstanding for address*.

Explanation: This is displayed if the debug option is used and the devices indicated have failed to close. This could occur as a result of the function *FindODB* not being able to find the I/O devices to shut down.

System Action: TCPIP continues.

User or Operator Response: Make sure that you have specified the correct physical or channel addresses of the I/O devices that you want to close.

System Programmer Response: Assist the operator as necessary.

Source Data Set: MVPCLOSE

Procedure Name: Mainline code

EZY1902I MVP Close Halt failed, *number I/Os for address* Halt Rc=*code*.

Explanation: This message is displayed if the debug option is used. This indicates that when an I/O failed to close, the *HALT* function was called to stop the indicated addresses but the device did not stop processing.

System Action: The I/O devices will fail to halt processing.

User or Operator Response: Stop the indicated devices using normal or abnormal shut down procedures and restart the devices if necessary.

System Programmer Response: Assist the operator if necessary.

Source Data Set: MVPCLOSE

Procedure Name: Mainline code

EZY1903I MVP Task did not close ddname *name*.

Explanation: This message displays if the debug option is used. The ddname indicated was not closed before the TCPIP task ended.

System Action: TCPIP continues with termination.

User or Operator Response: Make sure that the ddname is valid and that it exists in the volume defined to the MVS system. If you require help with this message, contact the IBM Software Support Center.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPUTIL

Procedure Name: DEBCHECK

EZY1904I MVP Application return routine *rc* to MVPTASK routine.

Explanation: This message displays if the debug option is used. MVPTASK was called to process the initial platform attached routine. The routine return code is displayed indicating a return from the task.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTASK

Procedure Name: MVPTASK

EZY1905I MVP Diagnose Number *DiagCode* issued:

Explanation: This message displays if the debug option is issued. It indicates the diagnose 7C subfunction code requested. This message should be followed by message EZY1906I, which specifies the registers allocated for this function.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDODG

Procedure Name: IssueDia

EZY1906I MVP Registers: *Regs1 Regs2 Regs3 Regs4*

Explanation: This message displays if the debug option is issued. It follows message EZY1905I or message EZY1929I. This message indicates the registers assigned for the functions described in either of the two preceding messages mentioned.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDODG

Procedure Name: IssueDia

EZY1907E MVP Diagnose *code* ignored, not supported yet.

Explanation: The module responsible for the initial handling of the Diagnose functions simulated by the MVS platform code received a request specifying a Diagnose function code that was not present in the simulation support. The incorrect Diagnose function code is identified in the message text. Control is returned to the calling module with a condition code of 0 and an error reason code of 0.

System Action: The invocation of MVPDODG is terminated immediately after issuing the message.

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

System Programmer Response: This error message is caused either by a programming error in the calling module or by corruption of the parameter list passed to the Diagnose simulation supervisor. Gather all available supporting documentation, and contact the IBM Software Support Center.

Source Data Set: MVPDODG

Procedure Name: Mainline code

EZY1910E MVP Diag7C subfunction *code* not supported.

Explanation: The module responsible for providing the simulation of the Logical Device Support Facility (LDSF) received an input parameter representing a Diagnose X'7C' subfunction code that was not valid. The message text identifies the incorrect subfunction code. The following subfunction codes are simulated by the MVS platform:

- Initiate (code 1)
- Accept (code 2)
- Present (code 3)
- Terminate (code 4)
- Terminate_All (code 5)
- Status (code 6)

EZY1912I • EZY1916I

- Break (code 1000)

See the appropriate *VM System Facilities for Programming* publication for information about the LDSF (under the subject area Diagnose X'7C'). The simulation provided by the MVS platform is used in conjunction with Telnet support. The subfunction code 1000 is an MVS-only code used to provide support for a line-mode Break.

System Action: Control is returned to the calling Telnet server module (TNLDSFP) with a condition code of 1 and an error reason code of 1. The invocation of MVPDG7C is terminated immediately. The Telnet server module subsequently fails the request it was processing at the time of the error.

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

System Programmer Response: This error message is caused either by a programming error in the associated Telnet server module or by corruption of the parameter list passed to the LDSF simulation routine. Gather all available supporting documentation, and contact the IBM Software Support Center.

Source Data Set: MVPDG7C

Procedure Name: Mainline code

EZY1912I MVP Diag7C Initiate-Accept followed by Status not supported.

Explanation: The module responsible for providing the simulation of the Logical Device Support Facility (LDSF) received an Initiate request with a parameter list specifying that an Initiate-Accept followed by a Status should be performed. This capability is not supported by the simulation facilities present in the MVS platform code.

System Action: The invocation of MVPDG7C is terminated immediately after issuing the message, and control is returned to the calling Telnet server module (TNLDSFP) with a condition code of 3 and an error reason code of 3. The Telnet server module subsequently fails the Initiate request that it was processing at the time of the error.

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

System Programmer Response: Because specification of this processing sequence depends on the value passed in the high-order byte of the second input parameter, a programming error in the calling module is the most probable cause. Corruption of the parameter list passed to the LDSF simulation routine is also a possible cause. Gather all available supporting documentation, and contact the IBM Software Support Center.

Source Data Set: MVPDG7C

Procedure Name: FINIT

EZY1913E MVP Diag7C Initiate-Logical Device specification not supported.

Explanation: The module responsible for providing the simulation of the Logical Device Support Facility (LDSF) received an Initiate request with a parameter list specifying that session initiation should be performed for a specific logical device address. This capability is not supported by the simulation facilities present in the MVS platform code.

System Action: The invocation of MVPDG7C is terminated immediately after issuing the message, and control is returned to the calling Telnet server module (TNLDSFP) with a condition code of 3 and an error reason code of 4. The Telnet server module subsequently fails the Initiate request that it was processing at the time of the error.

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

System Programmer Response: Because specification of this processing option depends on the value passed in the high-order byte of the second input parameter, a programming error in the calling module is the most probable cause. Corruption of the parameter list passed to the LDSF simulation routine is also a possible cause. Gather all available supporting documentation, and contact that IBM Software Support Center.

Source Data Set: MVPDG7C

Procedure Name: FINIT

EZY1916I MVP VTAM Request *request* not supported.

Explanation: This message displays if the debug option is used. The MVS VTAM interface received a request that is not supported. The command is ignored.

System Action: TCP/IP continues.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Find what is issuing the request and correct it. Contact the IBM Software Support Center if assistance is required.

Source Data Set: MVPVTWK

Procedure Name: MVPVTWK

EZY1917I MVP VTAM Send_Msg ignored, not supported.

Explanation: This message displays if the debug option is used. The MVS VTAM interface received a request for Send_Msg that is not supported. The request is not honored.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPVTWK

Procedure Name: MVPVTWK

EZY1921I MVP VTAM asked to initiate *token*, not implemented.

Explanation: This message displays if the debug option is used. The MVS VTAM interface received a request to initiate a session. This request is not supported by TCP/IP. The session ID or token requested is displayed.

System Action: TCPIP continues. If the token does not already exist, TCPIP sends a terminate request to VTAM for this token.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: Identify what is sending the initiate request from VTAM to TCP/IP and stop it. Contact the IBM Software Support Center for assistance if required.

Source Data Set: MVPVTWK

Procedure Name: MVPVTWK

EZY1926I MVP Received LDSF Ext Int: *ext_code*, Flag=LDSF_byte130 Reason=LDSF_reason.

Explanation: This message displays if the debug option is used. The MVS platform received an LDSF type external interrupt. The interrupt code, flag and reason code are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPIOINT

Procedure Name: ProcExt

EZY1927I MVP Attach of *module* failed with return code *rc*.

Explanation: The module responsible for attaching subtasks running under the MVS platform failed to attach the indicated module as a subtask of the platform. The return code from the ATTACH invocation is identified in the message text.

System Action: MVPATT posts the entry control block (ECB), for which MVPCALL is waiting, with a completion code indicating that the ATTACH failed. It then returns to a WAIT state, waiting on its list of ECBs. MVPCALL returns to its caller with a return code indicating that the requested subtask creation was unsuccessful. The calling module subsequently fails the underlying request.

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

System Programmer Response: See the applicable MVS *Application Development Reference: Services for Assembler*

Language Programs publication for information about the given ATTACH return code. If the error is because of an insufficient region size, make the appropriate adjustments, and restart the task. For return codes indicating logic errors, a programming error in either the MVS platform code or the module to be attached is probable. Gather all available supporting documentation, and contact the IBM Software Support Center. The following are the possible return codes and their descriptions:

Return Code	Description
0	Success
4	Was issued in a specify task abnormal exit (STAE)
6	No more storage
12	Invalid exit address or invalid parameter list address

Source Data Set: MVPATT

Procedure Name: Mainline code

EZY1928I MVP Attached *module* successfully.

Explanation: The indicated module was attached successfully by the MVS platform.

System Action: MVPATT posts the ECB, for which MVPCALL is waiting, with a successful completion code indicating that the ATTACH to the *module* was successful.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPATT

Procedure Name: Mainline code

EZY1929I MVP Diagnose Number *DiagCode* returned *CondCode* code

Explanation: This message displays if the debug option is issued. The requested diagnose code and a condition code are displayed. This message will be followed by message EZY1906I, which indicates the registers accessed for this function.

System Action: TCPIP continues.

User or Operator Response: Refer this to the system programmer.

System Programmer Response: Check the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information regarding the condition code received, and respond accordingly. Make sure the diagnose code number is valid.

Source Data Set: MVPDODG

Procedure Name: IssueDia

EZY1935I MVP Pwd Check failed *password/?????*, *rc* = *rc*, *rsn* = *reason*.

Explanation: This message displays if the debug option is used. The MVS platform called the SAF interface to check the password of a user ID. The password was not valid. Access is denied.

System Action: TCPIP continues.

User or Operator Response: Make sure that you used the correct syntax for the password for the indicated user ID. If you still have problems, it could be that the password is not set up accordingly. Refer this to your system administrator for correction.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPUTIL

Procedure Name: CHECKPWD

EZY1936I MVP *access req. access to data set name on vol for user id failed, Rc = rc, Rsn = code.*

Explanation: This message displays if the debug option is used. Access to the requested data set failed. Refer to *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about return codes.

System Action: TCPIP continues.

User or Operator Response: Respond as indicated by the return code. Reissue your access request.

System Programmer Response: Assist the user as required.

Source Data Set: MVPUTIL

Procedure Name: DEBUGMSG

EZY1937I MVP *system subsystem unavailable. system functions will fail.*

Explanation: During a scan of the Subsystem Communication Vector Table (SSCVT) control blocks, the platform code could not locate the SSCVT associated with the indicated subsystem.

System Action: The task is terminated, and control is returned to the system with a task completion code of 600. When the error occurs in module MVPVXI (implying a failure during the initialization of the Virtual Machine Communication Facility (VMCF) subsystem itself), subsystem initialization is terminated, and control is returned to the system with a completion code of 4000.

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

System Programmer Response: Check that the identified subsystem is created and initialized during system initialization. The subsystems should be defined to MVS using the SYS1.PARMLIB member IEFSSNxx. Refer to *ESA SPL: Initialization and Tuning* for information about this process. Make corrections to the definitions as necessary. If the definitions are correct, check for previous operator messages that might indicate why the particular subsystem failed to initialize. See the appropriate *System Messages* book for information about possible error messages from subsystem processing. A programming error is the most probable cause of the problem if subsystem processing error messages occur. If the problem cannot be remedied by modifications to subsystem definitions, gather all available documentation, and contact the IBM Software Support Center.

Source Data Set: MVPMAIN, MVPXVI

Procedure Name: LOCCVT

EZY1940I MVP VMCF Function *function invoked.*

Explanation: This message displays if the debug option is used. It indicates the virtual machine communication facility (VMCF) function passed to TCPIP.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDOVM

Procedure Name: SENDVMCF

EZY1944I MVP **Help** *message*

Explanation: This message indicates that the help facility has been invoked for the "Modify" command. The following list displays the different subcommands accepted for the Modify command:

Command	Description
---------	-------------

WakeVTAM	Posts VTAM Q.
----------	---------------

Dump	Cause program interrupt in MVPOCM, which gets a dump and terminates TCPIP.
------	--

EZY1947I • EZY1948I

StopVTAM

Queues STOP to VTAM Interface task.

LUact Activate or reactivate a logical unit to TCPIP.

LUinact

Deactivate a logical unit to TCPIP.

LUstat Show logical status to TCPIP.

Debug Displays all switches and their state.

Debug *switch-name* (On, Off, or Display).

Note: The debug command is used to turn switches on or off, or display its attributes.

DfltAppl

Change Transparent default application name.

LineAppl

Change default line mode application name.

DmpReset

Allow another dump to be taken in case of error.

DumpIsaQ

Print out MVPISAQ areas.

DumpVtbl

Print out VTAM Application, LU, IP, and mapping tables.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1947I MVP Invalid command *command*.

Explanation: The MVS platform received a command for TCPIP that was not valid. See message EZY1944I for a list of valid commands and their descriptions.

System Action: TCPIP continues.

User or Operator Response: Reissue your request using a valid command.

System Programmer Response: Assist the operator if required.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1948I MVP Default Application is now *name*.

Explanation: The new default application name (for either transparent or line mode) is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1950I MVP IUCV Query: Rc=rc IpRcode=code IpAudit=name.

Explanation: This message displays if the debug option is issued. An inter-user communication vehicle (IUCV) function was called. A system return code, IP return code, and IP audit address are displayed in this departing message from an IUCV function.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the IP return code for help in resolving any errors. A list of IP error codes and system return codes and their descriptions can be found in the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MVPDOIUC

Procedure Name: Depart

EZY1951I MVP VMCF terminated: Cannot function without TNF.

Explanation: During the Virtual Machine Communication Facility (VMCF) address space initialization processing, a scan of the Subsystem Communication Vector Table (SSCVT) control blocks was performed, but the code could not locate the SSCVT associated with the Termination Notification Facility (TNF) subsystem. (The VMCF and TNF subsystems provide simulations of facilities provided by the VM operating system for the MVS platform code.)

System Action: The VMCF subsystem initialization is terminated and control is returned to the system with a completion code of 5000.

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

System Programmer Response: Check that the Termination Notification Facility (TNF) subsystem is created and initialized during system initialization. The subsystem should be defined to MVS using the SYS1.PARMLIB member IEFSSNxx. Refer to *z/OS Communications Server: IP Configuration Reference* and *ESA SPL: Initialization and Tuning* for information about this process. Make corrections to the definitions, as necessary. If the definitions are believed to be correct, check for previous operator messages that might indicate why the TNF subsystem failed to initialize. See the appropriate *System Messages* book for information about possible error messages from subsystem processing. A programming error is the most probable cause of the problem if subsystem processing error messages occur. If the problem cannot be remedied by modifications to subsystem definitions, gather all available documentation, and contact the IBM Software Support Center.

Source Data Set: MVPXVI

Procedure Name: Mainline code

EZY1952I MVP SSI: IEFSSREQ Process Sysout failure, rc=rc.

Explanation: This message displays if the debug option is used. The subsystem request process failed as indicated by *rc*.

In the message text:

rc the *rc* value, minus a decimal value of 100, is the JES SSI return code.

System Action: TCPIP continues.

User or Operator Response: Refer any problems to the system programmer.

System Programmer Response: This message is used for debugging purposes only. For an explanation of the JES SSI return code, see the return code information for SSI Function Code 1 in the *z/OS MVS Using the Subsystem Interface*. Refer any questions to the IBM Software Support Center.

Source Data Set: MVPSSI

Procedure Name: MVPSSI1

EZY1953I **MVP SSI: SSOBRETN Process Sysout failure, rc=rc.**

Explanation: This message displays if the debug option is used. The subsystem interface failed as indicated by *rc*.
In the message text:

rc the *rc* value, minus a decimal value of 1000, is the JES SSOB return code.

System Action: TCPIP continues.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: This message is used for debugging purposes only. For an explanation of the JES SSOB return code, see the SSOBRETN return codes for SSI Function Code 1 in the *z/OS MVS Using the Subsystem Interface*. Refer any questions to the IBM Software Support Center.

Source Data Set: MVPSSI

Procedure Name: MVPSSI1

EZY1954I **MVP SSI: Process Sysout rc=rc, no data sets to select.**

Explanation: This message displays if the debug option is used. The subsystem interface request could not complete because no data sets were found to select.

In the message text:

rc is the JES SSOB return code SSSOEODS (no more data sets to select).

System Action: TCPIP continues.

User or Operator Response: Make sure that you define a job data set for the subsystem interface to process. Make sure that you follow correct data set naming conventions as described in *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the operator if required.

Source Data Set: MVPSSI

Procedure Name: MVPSSI1

EZY1955I **MVP SSI: Process Sysout rc=rc, Job not found.**

Explanation: This message displays if the debug option is used. The subsystem interface could not locate a specified job name.

In the message text:

rc is the JES SSOB return code SSSONJOB (job not found).

System Action: TCPIP continues.

User or Operator Response: Make sure you define a correct job name to the subsystem interface routine for SMTP.

System Programmer Response: Assist the operator if required.

Source Data Set: MVPSSI

Procedure Name: MVPSSI1

EZY1956I **MVP SSI: Selected Job: number # job id.**

Explanation: This message displays when the debug option is used. The MVS platform subsystem interface called the routine that gets a job and allocates a data set for SMTP. The indicated job number and ID for the obtained job are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPSSI

Procedure Name: MVPSSI1

EZY1957I MVP SSI: DYNALLOC Return code(s) rcode(s).

Explanation: This message displays if the debug option is used. The MVS subsystem interface routine was called to dynamically allocate a data set. The indicated return codes are displayed. Any return code other than 0 indicates an error in allocating the data set.

System Action: TCPIP continues.

User or Operator Response: Refer any errors to the system programmer.

System Programmer Response: Data set naming conventions are normally the main cause of errors in the dynamic allocation routine. Refer to *z/OS Communications Server: IP Configuration Reference* for more information about dynamically allocating data sets.

Source Data Set: MVPSSI

Procedure Name: MVPSSI1

EZY1958I MVP SSI: Allocate Job return rc job ddname.

Explanation: This message displays if the debug option is used. The MVS platform subsystem interface was invoked to allocate a call. The job number, ddname, and return code are displayed. If the return code is nonzero, the call is not allocated.

System Action: TCPIP continues.

User or Operator Response: Refer to the system programmer if the return code is nonzero.

System Programmer Response: If the return code is 8, this could indicate that the job number, ddname, or both are not valid. Check the job number or ddname and make sure that they are identified correctly to the subsystem interface. Correct the error and reinitiate the subsystem interface. If you continue to have problems with this message, contact the IBM Software Support Center.

Source Data Set: MVPSSI

Procedure Name: MVPSSI2

EZY1959I MVP SSI: Free Job return rc job action.

Explanation: This message displays if the debug option is used. The MVS platform subsystem interface was invoked to free a call. The job number, action requested, and return code are displayed. If the return code is nonzero the call is not freed.

System Action: TCPIP continues.

User or Operator Response: Refer to the system programmer if the return code is nonzero.

System Programmer Response: If the return code is 8, this could indicate that the action requested, job number, or both are not valid. Correct the error and reinitiate the subsystem interface. If you continue to have problems with this message, contact the IBM Software Support Center.

Source Data Set: MVPSSI

Procedure Name: MVPSSI3

EZY1960I MVP SSI: MVPSSIW, state received.

Explanation: This message displays if the debug option is used. The MVS subsystem interface that waits for a JES file was called. The *state* of the call could be **Enable**, which specifies to look for a file or **Disable**, which specifies not to look for a file. If the state indicated is **Bad Post**, then an error occurred while posting the event control block (ECB).

System Action: TCPIP continues.

User or Operator Response: Refer any problems to the system programmer.

EZY1961I • EZY1964I

System Programmer Response: If the message indicates a bad post, you will need to gather all source information and any dumps that may be required and contact the IBM Software Support Center for help.

Source Data Set: MVPSSIW

Procedure Name: MVPSSIW

EZY1961I MVP SSI: MVPSSIW, Got Job *job* to handle.

Explanation: This message displays if the debug option is used. The MVS platform subsystem interface got the job indicated and will continue processing.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPSSIW

Procedure Name: MVPSSIW

EZY1962I MVP SSI: MVPSSIW, No Job to handle.

Explanation: This message displays if the debug option is used. The MVP platform subsystem interface was called to determine if there were any jobs to process, but no jobs were found.

System Action: TCPIP continues.

User or Operator Response: Normally, this message does not reflect an error situation, but only indicates that there are no jobs to process. If errors occurred during subsystem interface processing, messages EZY1952I–EZY1955I may have been issued to report the error. If there are jobs to process, make sure that you have identified a correct job number to the subsystem interface. Reinitiate the interface after correcting the problem.

System Programmer Response: Assist the user as required.

Source Data Set: MVPSSIW

Procedure Name: MVPSSIW

EZY1963I MVP SSI: Query called *jobid* Rcode=*queryrc*. (or) Allocate called *jobid* *alocddn* Rcode=*allocrc*. (or) Free called *jobid* *action* Rcode=*freerc*.

Explanation: This message displays if the debug option is used. The RDRQUERY, RDRALLOC, or the RDRFREE function is called to select, allocate, or free a job's data set for the subsystem interface. A job ID and a return code are returned for each function. The data set name is displayed when the allocate function is called, and an *action* indicating whether the data set has been freed is also displayed.

System Action: TCPIP continues.

User or Operator Response: Refer any errors to the system programmer.

System Programmer Response: Refer to the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for information about return codes, and respond accordingly.

Source Data Set: MVPDOSI

Procedure Name: RDRQUERY, RDRALLOC, RDRFREE

EZY1964I MVP SSI: Interrupt called 0 (or) 1 Rcode=*rc*.

Explanation: This message is issued when the debug option is used. The RDRINT routine was called. This routine turns on or off the ability for the subsystem interface to accept interrupts from the MVS platform. Any return code other than 0 indicates a problem with this routine.

System Action: TCPIP continues.

User or Operator Response: Refer any errors to the system programmer.

System Programmer Response: Refer to the *z/OS Communications Server: IP Sockets Application Programming Interface*

Guide and Reference for information about return codes. Respond to any bad return codes as indicated.

Source Data Set: MVPDOSI

Procedure Name: RDRINT

EZY1965I MVP SSI: MVPSSIW, wait for file task started | ended.

Explanation: This message displays if the debug option is used. This messages marks the start or the end of the MVS subsystem interface that waits for a JES file.

System Action: Task ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPSSIW

Procedure Name: MVPSSIW

EZY1966I MVP Racf R15 = rc, SAFPRRET = rc, SAFPRREA = reason.

Explanation: This message displays if the debug option is used. The MVS system called the SAF interface to define a resource profile for a user ID and data set combination. A return code, system return code, and reason code are displayed.

System Action: TCPIP continues.

User or Operator Response: Respond as indicated by the return and reason codes. For more information about return codes, refer to *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*. For more information about RACF see *z/OS Security Server RACROUTE Macro Reference*. If you need additional help with this message, call the IBM Software Support Center.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPUTIL

Procedure Name: CHECKDSN

EZY1967I MVP RACSTAT Rc=rc.

Explanation: This message displays if the debug option is used. The MVS platform called the SAF interface to check if a security product is active. A return code of 0 indicates that the product is active. A return code other than 0 indicates that the product is not active.

System Action: TCPIP continues.

User or Operator Response: If the security product should be active and it is not, refer this to the system programmer.

System Programmer Response: Start the security product or perform an analysis on the product for information about why the product is not active.

Source Data Set: MVPUTIL

Procedure Name: DEBUGMSG

EZY1968I MVPAUTH: Getmain Rc=code

Explanation: This message is displayed if the debug option is used. The function *Getmain* which gets storage for the allocation of data sets has returned with the indicated return code, specifying that storage is not available. The data set function could not be processed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: See *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG* for any help in resolving return codes. None.

EZY1969I • EZY1973I

Source Data Set: MVPAUTH

Procedure Name: Internal_Auth_Check

EZY1969I MVPAUTH: Freemain Rc=code

Explanation: This message is displayed if the debug option is used and the indicated data set in message EZY1970I could not be located. Storage may or may not have been freed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the return code. For information about MVS return codes affecting the Freemain function, see *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG*.

Source Data Set: MVPAUTH

Procedure Name: Internal_Auth_Check

EZY1970I MVPAUTH: DSN=data set name Locate Rc=code

Explanation: This message is displayed if the debug option is used and the indicated data set could not be located. If the return code is zero, then the indicated data set has been located.

System Action: TCPIP continues.

User or Operator Response: Check the syntax of the data set requested and make sure that it is a valid data set. Reissue your request with the correct data set name.

System Programmer Response: None.

Source Data Set: MVPAUTH

Procedure Name: Internal_Auth_Check

EZY1972I MVP VMCF Function function: Rc=return code

Explanation: This message displays if the debug option is used. Displayed is the VMCF function called by TCP/IP and the resulting return code.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDOVM

Procedure Name: SENDVMCF

EZY1973I MVP IUCV: function invoked.

Explanation: This message displays if the debug option is issued. An inter-user communication vehicle (IUCV) function was called.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDOIUC

Procedure Name: Welcome

EZY1974I **MVP Issued Command:** *Rc=rc R0=r0 "command text"*.

Explanation: This message displays if the debug option is used. The MVS system received the indicated command. A return code and register 0 data (address space ID) are returned. If the return code is not zero, the command failed for START command.

System Action: TCPIP continues.

User or Operator Response: If the return code is not zero, check the syntax of the command issued and make sure it is correct. Make sure that the command is valid for the MVS system. If you continue to experience problems with this message, contact the IBM Software Support Center.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPUTIL

Procedure Name: Do_Command

EZY1975I **Attached task at address** *address* **starting.**

Explanation: This message displays if the debug option is used. The asynchronous caller routine was called to attach a task. This message indicates that the asynchronous caller routine will attach a task for the routine with entry point of address to run under.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPCCALI

Procedure Name: MVPCCALI

EZY1976I *load module name* **cannot Stop until started, try later.**

Explanation: This message is issued if the debug option is used. The MVS platform received a command to stop the indicated load module, but it has not been started.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: MVPOCM1

EZY1977I **TaskExist rc task rc for user** *user id* **Asid** *taskasid*

Explanation: This message displays if the debug option is used. The routine to check if a task exists is called to check if the requested task is available. The return code, user ID, and address space ID are displayed.

System Action: TCPIP continues.

User or Operator Response: Refer any errors to the system programmer.

System Programmer Response: Refer to the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for any information about error codes, and respond accordingly.

Source Data Set: MVPDOSY

Procedure Name: TASKEXIS

EZY1978I MVP Error setting Ext Interrupt STIMER(M) rc=*rc*

Explanation: This message displays if the debug option is used. The MVS platform received an error setting up an external interrupt or the interrupt type is not valid.

System Action: TCPIP continues.

User or Operator Response: Refer any problems to the system programmer.

System Programmer Response: The following table shows the possible return codes and their descriptions:

Code	Description
4	Request completed with problems.
8	Request could not be performed.
12	Error detected in an input parameter.
16	A system error occurred while performing the requested function.
99	Invalid interrupt type.

Respond to the return code as indicated. Call the IBM Software Support Center for additional help with this error.

Source Data Set: MVPEXTR

Procedure Name: MVPEXTR

EZY1980E *module* terminating due to previous error(s).

Explanation: TCPIP initialization failed due to unrecoverable errors. This message is issued when either the Termination Notification Facility (TNF) or Virtual Machine Communication Facility (VMCF) TNF or VMCF subsystems are not running. If this message is preceded by message number EZY1870 then this message displays because the MVPTASK function was not loaded. If message number EZY1981 precedes this message then there was insufficient storage available. See the respective message numbers for more information.

System Action: TCPIP terminates.

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

System Programmer Response: Make sure that there is an entry in the IEFSSNxx PARMLIB member for TNF and VMCF. If TNF and VMCF are running, include a MORETRACE TCPIP statement in the *hlq*.TCPIP.PROFILE and restart. After determining the error, check SYSDEBUG, SYSERR, SYSERROR, and the *hlq*.PROFILE.ERROR data sets for diagnostic information. You should also check and confirm that the module is located in the library or in the case of insufficient storage, find out why there is not enough storage and increase storage if necessary.

Source Data Set: MVPMAIN

Procedure Name: Mainline code

EZY1981I *task* insufficient storage.

Explanation: An attempt to obtain virtual storage in Subpool 73 for I/O completion and attention work areas was unsuccessful.

System Action: The *task* is terminated and control is returned to the system with a task completion code of 8xx, with the xx portion reflecting the GETMAIN return code.

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

System Programmer Response: Determine if an increase in the region size of the started task will alleviate the problem. If the region size is not the problem, the most probable cause is a programming error involving a failure to release storage. Gather all available documentation and report the error to the IBM Software Support Center.

Source Data Set: MVPMAIN

Procedure Name: Mainline code

EZY1982E Terminating: one *task* is already active.

Explanation: To prevent the initiation of multiple TCPIP address spaces for the same task, the MVS platform code uses the ENQ macro to serialize the use of TCPIP resources. At task initialization, MVPMAIN issues an ENQ macro with a resource name of TCPIPSYS.task_name, requesting exclusive use. The message indicates that the ENQ failed.

System Action: The task is terminated and control is returned to the system with a task completion code of 9xx, with the xx portion reflecting the ENQ return code.

User or Operator Response: If the error was an unintentional attempt to start a TCPIP task that was already active, no actions are required. Otherwise, tell the system programmer about the error.

System Programmer Response: If the error was not because of an unintentional attempt to start a duplicate TCPIP task, examine the task completion code provided when the task was terminated. It will be of the form 9xx, with the xx portion of the code being the ENQ return code. See the appropriate *Application Development Reference: Services for Assembler Language Programs* publication for information about interpreting the return code. Follow the recommended problem resolution procedures indicated by the appropriate *MVS Diagnosis* book.

Note: The ENQ was issued with the RET=USE parameter.

If problem determination indicates that the MVS platform is in error, gather all available documentation and report the error to the IBM Software Support Center.

Source Data Set: MVPMAIN

Procedure Name: Mainline code

EZY1984I DIAG98: UnLock Page: Virtual=address

Explanation: This message displays if the debug option is used. The MVS system called the DIAG98 subroutine to unlock a page in central storage. The virtual address is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPUTIL

Procedure Name: MVPUTIL

EZY1985I DIAG98: DoDiag98: Device=address CCWP=virtual channel control word CCWreal=real channel control word

Explanation: This message displays if the debug option is issued. The DoDiag98 subroutine of the Diag7C routine was called to perform I/O functions for a device with real channel control words (CCWs) that have been translated to virtual CCWs. The device address, virtual CCWs, and real CCWs are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPDOIO

Procedure Name: DoDiag98

EZY1986I DIAG98: PgFix: Vaddr=address CC=code

Explanation: This message displays if the debug option is used. The DIAG98 subroutine was called to fix a page in the MVS system. The virtual address and completion code are returned.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

EZY1987I • EZY1991I

Source Data Set: MVPUTIL

Procedure Name: PgFix

EZY1987I **DIAG98: PgFree: Vaddr=address CC=code**

Explanation: This message displays if the debug option is used. The DIAG98 subroutine was called to free a previously fixed page of storage in the MVS system. The virtual address and completion code are provided.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPUTIL

Procedure Name: PgFree

EZY1988I **DIAG98: UnLock All: FXBaddr=address FXBnext=address Vaddr=address**

Explanation: This message displays if the debug option is used. The MVS platform system called the diagnose 98 subroutine to free a whole chain of virtual address. The current fixed page, next fixed page, and the virtual address are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPUTIL

Procedure Name: MVPUTIL

EZY1989I **MVP Logical Unit *name* is now active to TCP/IP.**

Explanation: The displayed logical unit is now active.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1990I **MVP Logical Unit *name* is now inactive to TCP/IP.**

Explanation: This message displays in response to your command to deactivate a logical unit to TCPIP. The LU is now inactive.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1991I **MVP Logical Unit *name* is inactive to TCP/IP.**

Explanation: This message displays in response to a status request. The LU name specified is not active to TCPIP.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1992I MVP Logical Unit *name* is active to TCP/IP and in use.

Explanation: This message displays a response to a status request. The LU name indicated is currently active and in use.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1993I MVP Logical Unit *name* is active to TCP/IP and not in use.

Explanation: This message displays in response to a status request. The LU name specified is active to TCPIP but currently not in use.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1994E MVP Logical Unit *name* not inactivated – in use to TCP/IP. User must issue LUINACT FORCE, luname to deactivate.

Explanation: This message indicates that the MVS platform received a request to deactivate an LU to TCPIP that failed. The LU remains active.

System Action: TCPIP continues.

User or Operator Response: Make sure that the LU is no longer in use and, as the message indicates, you must issue "LUINACT FORCE" followed by the LU name to deactivate it.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY1995E MVPEXCP: MVPMAIN has Diag98 support but *load module* does not.

Explanation: This message displays if the debug option is used. The mainline code (MVPMAIN) has support for Diag98, or real channel control words (CCWs), but the PC load module indicated does not. Messages EZY1996E and EZY1997E will follow this message with more information.

System Action: TCPIP continues.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Verify that the correct version of the load module is being used. Call the IBM Software Support Center for help if necessary.

Source Data Set: MVPEXCP

Procedure Name: MVPEXCP

EZY1996E Continuing using VCCWS (virtual CCWs) option.

Explanation: This message should be preceded by message EZY1995E, which indicates that a conflict exists between the load module and the mainline code (MVPMAIN) regarding support for the Diag98 function. The installation will continue, but it will use virtual channel control words (CCWs) instead of real CCWs.

System Action: TCPIP continues.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: See message EZY1995E for help with the message.

Source Data Set: MVPDODG

Procedure Name: Query98

EZY1997E Installation was not done correctly.

Explanation: Your installation was not done correctly. See messages EZY1995E and EZY1996E, which precede this message for more information.

System Action: TCPIP continues.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: See previous messages and their descriptions.

Source Data Set: MVPDODG

Procedure Name: QAmode, Query98

EZY1998E DIAG98: MVPMAIN does not support real CCWs but *load module name* does.

Explanation: The mainline code (MVPMAIN) does not have support for Diag98, or real channel control words (CCWs), although the load module indicated does. Messages EZY1996E and EZY1997E should follow this message.

System Action: TCPIP continues.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: Check the mainline code (MVPMAIN) and make sure that it is running the code with support for Diag98 as defined in this source module (MVPDODG). Contact the IBM Software Support Center if you need more help with this message.

Source Data Set: MVPDODG

Procedure Name: Query98

EZY1999E QAmode: MVPMAIN and *load module name* are not at the same level.

Explanation: This message displays if the MVPMAIN code, which functions as a mainline entry for the MVS platform, does not have the same level of code for QAMODE support as the load module displayed. This message precedes message EZY1997E, which indicates that your installation was not done correctly.

System Action: TCPIP continues.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: Check the mainline code (MVPMAIN) and make sure that it is running the code with support for QAMODE as defined in this source module (MVPDODG). Contact the IBM Software Support Center if you need more help with this error.

Source Data Set: MVPDODG

Procedure Name: QAmode

Chapter 3. EZY2xxxx messages

EZY2000I OCM General Default Application not configured for Transparent Mode.

Explanation: This indicates that the request to change the default application could not be honored because the new application is not configured for Transparent mode.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY2001I OCM General Default Application not configured for Line mode.

Explanation: This indicates that the request to change the general default application could not be honored because the new application is not configured for line mode.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: AnyL

EZY2016I MVP Diag7C LocAppl: Unknown function request *function* for ApplName *application*

Explanation: The *locate application* subroutine of the Diagnose 7C routine has returned with an unknown function request for the indicated application. The *LocAppl* subroutine checks the application request against the application table, declared in the MVPAPPL source code data set and defined in your *hlq.PROFILE.TCPIP* data set, to determine if the function request is allowed for the specified application.

System Action: TCPIP continues. The specified request is not processed.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: The function flags that are passed for these applications are either *restricted* (1) or *disconnectable* (2). Make sure that the function specified for the requested application is defined in your *hlq.PROFILE.TCPIP* data set. Reissue your request specifying a valid function for the application requested.

Source Data Set: MVPDG7D

Procedure Name: LocAppl

EZY2032W LU name *lu name* specified in the *command* command does not match defined LU name or LU group name.

Explanation: The LU name or LU group entered in the command was not found in the LU status tables.

System Action: The command is not performed. TCPIP continues.

User or Operator Response: Correct the LU name and resubmit the command.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2034W LU specified in *command* is already inactivated.

Explanation: The LU name specified for the LUINACT command was already inactivated.

System Action: The command is not performed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2035W Modify command *command* Incomplete. It is ignored.

Explanation: The required keyword was not specified.

System Action: The command is not performed. TCPIP continues.

User or Operator Response: Refer to *z/OS Communications Server: IP Configuration Reference* for more information about the MODIFY command.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2036I Modify command DEBUG *debug state* is accepted.

Explanation: A change to the DEBUG flag has been changed to the specified value. The new debug option setting is now in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2037W Invalid modify command. DEBUG *debug state* is ignored.

Explanation: An incorrect DEBUG option has been specified as part of the MODIFY command. The Telnet server ignored the DEBUG request. The debug option is not changed.

System Action: TCPIP continues.

User or Operator Response: Check the DEBUG option specified, correct, and resubmit the MODIFY command.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2038I Modify *command* LU specified matches LUGROUP *lu*. Accepted.

Explanation: The modify command was accepted because the specified LU was found in the LU group table.

System Action: The command is performed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2039W LU specified *lu* is already activated. Ignored.

Explanation: The LU name specified with the LUACT parameter was already activated.

System Action: The command is not performed. TCPIP continues.

User or Operator Response: Check the specified LU and if incorrect, correct and reissue the MODIFY command.

System Programmer Response: Assist operator as required.

Source Data Set: MVPOCM

Procedure Name: ANYL

EZY2040I SMSG: VMCF is not active on the system.

Explanation: The message indicates that the program attempted to set up a communication session using the Virtual Machine Communication Facility (VMCF). The system was unable to locate or save the address space of the VMCF communication vector table. The program failed to initialize the client interface to VMCF.

System Action: The communication does not occur because VMCF address space could not be established. The program requires the address space to continue. The program sets a return code of 16, indicating a system error to the calling program, and terminates execution. The VMCF address display command is not issued.

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

System Programmer Response: Make sure that the communication vector table and VMCF are active.

Source Data Set: MVPXDISP, MVPXSMSG

Procedure Name: Mainline code

EZY2041I SMSG: Userid not specified.

Explanation: TCPIP received an SMSG command for the VMCF interface. You did not specify the user program, address space, or batch job to receive the SMSG.

System Action: TCPIP continues.

User or Operator Response: Reissue the SMSG command specifying a user program, address space, or batch job. For more information about SMSG refer to the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*, or *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2042I SMSG: Userid specified incorrectly.

Explanation: The MVS platform received an SMSG command for the VMCF interface. You specified a user program, address space, or batch job that was too long.

System Action: TCPIP continues. The SMSG command is not honored.

User or Operator Response: Reissue the SMSG command specifying a user program, address space, or batch job of the correct length. For more information about SMSG refer to the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*, *z/OS Communications Server: IP User's Guide and Commands*, or *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2043I SMSG: No message specified.

Explanation: TCPIP received an SMSG command with no message specified. The SMSG command is used to send a special message (SMSG) to a user program, address space, or batch job. The message text is required.

System Action: TCPIP continues. The SMSG command is not honored.

User or Operator Response: Reissue the SMSG command specifying the text of the SMSG to be sent. If this message occurred as a result of issuing SMSG TRACE on TSO from where the FTP server resides, then you should issue "SMSG FTPSERVE TRACE". For more information about SMSG, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*. If you need more help with this message, contact the IBM Software Support Center.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2045I SMSG: user not logged on.

Explanation: TCPIP received an SMSG command for a user that is not logged on.

System Action: TCPIP continues. The SMSG command is not honored.

User or Operator Response: Check and make sure the target is active and reissue the SMSG command. For more information about the SMSG command, refer to *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2046I SMSG: user not receiving; SMSG off.

Explanation: TCPIP received an SMSG command for a user. The user is not receiving because the SMSG interface is not set on.

System Action: TCPIP continues. The SMSG interface is off.

User or Operator Response: Use the SETSMSG command to set the SMSG interface on for the target user. For more information refer to *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2047I SMSG: VMCF return code rc, SMSG failed.

Explanation: TCPIP received a bad return code from the VMCF function related to a special message (SMSG) request. The special message (SMSG) command is not performed.

System Action: TCPIP continues.

User or Operator Response: When TCP/IP establishes a session with VMCF, it does not enable the SMSG interface. If your program accepts SMSG commands, you must enable the reception of SMSGs before using the services provided by the application program. The SETSMSG command is used to enable and disable the reception of SMSGs. For more information about the SMSG and SETSMSG commands for VMCF applications, refer to *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*, or *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2048I SMSG: *user* quiesced; SMSG ignored.

Explanation: The SMSG command sent to the indicated user is ignored because the user or program is not active.

System Action: TCPIP continues.

User or Operator Response: Make sure that the user or program is active and reissue the SMSG command. Contact your system administrator for help if required.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2049I SETSMSG: Bug in SETSMSG command, invalid code in VMCF call.

Explanation: The SETSMSG command cannot be honored. The code in the VMCF call is not valid.

System Action: TCPIP continues. The SETSMSG command is not honored.

User or Operator Response: Refer this problem to the system programmer.

System Programmer Response: Contact the IBM Software Support Center for assistance with this message.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2050I SETSMSG: Specify On or Off.

Explanation: The SETSMSG command was received but you did not specify "ON" or "OFF".

System Action: TCPIP continues. The SETSMSG command is not honored.

User or Operator Response: Reissue the SETSMSG followed with "ON" or "OFF". For more information about the SETSMSG command, refer to *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: Assist the operator as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2051I SETSMSG: You are not authorized to set *user's* flag.

Explanation: The SETSMSG command was received for the indicated user or program. This message indicates that you are not authorized to issue this command for the user or program. To be honored, the SETSMSG command must be issued by the affected user or program.

System Action: TCPIP continues. The SETSMSG command is not honored.

User or Operator Response: Contact the user or owner of the program to have them issue the SETSMSG command.

System Programmer Response: Assist the user as required.

Source Data Set: MVPXSMSG

Procedure Name: MVPXSMSG

EZY2053I MVPXDISP: User *userID* Asid *addrspaceid*.

Explanation: This message is the first in a set of messages displaying information about a specific user ID. The user ID *userID* indicates the client ID, and the Address Space Identifier (asid) for that client is displayed.

System Action: The system continues with message EZY2054I, displaying user information for this ID.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: Mainline code

EZY2054I MVPXDISP: Data @ *user data address* **Sm**=*system mask* **Cr0**=*control register 0* **Flags**=*flags*.

Explanation: This message displays in response to the MVPXDISP command after message EZY2053I. The user data information is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2055I MVPXDISP: Client of *interface*.

Explanation: This message displays in response to the MVPXDISP command and follows message EZY2054I. This message can display up to four times, once for each possible interface. The following list displays the four possible interfaces that can display:

- VMCF address space
- SMSG
- VMCF
- IUCV

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2056I MVPXDISP: IUCV mask=*mask*, **Pending Ctl**=*control*, **Appl**=*application*.

Explanation: This message displays in response to the MVPXDISP command and follows message EZY2055I. Information about the IUCV application is displayed. The information displayed includes enable, control pending interrupts, and application pending interrupts masks for the client.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2057I MVPXDISP: VMCF: Buf=*address*, **Len**=*length*, **Flgs**=*flags* **user**=*user* **Key**=*auth key*.

Explanation: This message displays in response to the MVPXDISP command. The user data is displayed. Included are the user external interrupt buffer address and length, user flags, and user authority.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2058I **MVPXDISP: IUCV: Connections=number, Max=number.**

Explanation: This message displays in response to the MVPXDISP command and follows message EZY2059I. The number of IUCV connections and the maximum number of IUCV connections allocated for the client are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2059I **MVPXDISP: VMCF: Pending count=number Flags=flags.**

Explanation: This message displays in response to the MVPXDISP command and follows message EZY2057I. The number of VMCF pending connections and the VMCF flags are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2060I **MVPISAQ : name1 name2 Header at address**

Explanation: This message displays when the debug option is used. The header name and address are displayed for an initial storage area (ISA) queue header. This message displays with messages EZY2061I — EZY2063I.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: PRISAQ

EZY2061I **MVPISAQ : Subpool number 1st Getmain count count**

Explanation: This message displays when the debug option is used. The subpool number and the total number of storage areas allocated by GETMAIN are displayed. This message displays with messages EZY2060I, EZY2062I, and EZY2063I.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: PRISAQ

EZY2062I **MVPISAQ : 2nd Getmain count count**

Explanation: This message displays when the debug option is used. The count of overflow ISA areas allocated is displayed. This message displays with messages EZY2060I, EZY2061I, and EZY2063I.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

EZY2063I • EZY2066I

Source Data Set: MVPOCM

Procedure Name: PRISAQ

EZY2063I **MVPISAQ : Frame size** *size* **Max asked** *max size(max size)*

Explanation: This message displays when the debug option is used. The default frame size and the largest storage area allocated are displayed in decimal and hexadecimal form. This message displays with messages EZY2060I — EZY2062I.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: PRISAQ

EZY2064I **MVPXDISP: User** *name* **not found.**

Explanation: This message displays in response to the MVPXDISP command. The user indicated could not be found by this utility.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2065I **MVPXDISP: IUCV: Ctl flags=***flags* **Appl flags=***flags*.

Explanation: This message displays in response to the MVPXDISP command and follows message EZY2058I. The IUCV control flags and the IUCV application flags are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2066I **MVPXDISP: Index=***asid*, **Sm=***system mask*, **Cr0=***data*, **Flags=***flags*, **User=***user*.

Explanation: This message displays in response to the MVPXDISP command. VTAM user information is displayed, including the address space ID, system mask, control register 0 data, flags, and user name.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXDISP

Procedure Name: MVPXDISP

EZY2080I **MVPMAIN: Storage could not be obtained for TCP/IP address CVT, Name=*pc load module name*, Rcode=*rc***

Explanation: This message indicates that there is not enough common storage available for the TCPIP communication vector table (CVT) control block. This will not directly impact TCPIP functions. Some cleanup may not be performed when TCPIP is stopped.

System Action: TCPIP continues.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Investigate why below the line, CSA storage is not available. Restart TCPIP when more common storage becomes available. For more help contact the IBM Software Support Center.

Source Data Set: MVPMAIN

Procedure Name: mainline

EZY2081I **MVPMAIN: Anchor control block could not be found for TCP/IP address space CVT**

Explanation: The TCPIP address space CVT could not be located because the control block containing its address could not be found.

System Action: TCPIP continues.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Look for abend in GETMAIN during TNF subsystem initialization. Correct that problem then restart TCPIP after you IPL the MVS system. For more help contact the IBM Software Support Center.

Source Data Set: MVPMAIN

Procedure Name: mainline

EZY2090I **DUMPING VTAMinfo tables, Table number = *table***

Explanation: Information notifying you that TCPIP is about to display the indicated VTAM tables. The following are the application tables used by VTAM:

Table Number	Description
0	All application tables
1	Telnet application table
2	VTAM log mode information table
3	LU group table
4	IP group table
5	IP to LU map table
6	Default application table
7	Application table
8	LU status table (for all LUs)

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2091I **—**

Explanation: Message separator.

System Action: The program continues.

User or Operator Response: None.

EZY2092I • EZY2096I

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2092I **TELNET** *id=applid*

Explanation: Information notifying you that TCPIP is about to display the Telnet application table used by VTAM.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2093I **Mode Table**

Explanation: Information notifying you that TCPIP is about to display the mode table used by VTAM.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2094I **Mt Ptr=table, ElemCount=counter**

Explanation: Information specifying the allocated storage, pointed to in the *table* value, for the VTAM Telnet mode table. The *counter* value specifies the number of entries in the mode table.

System Action: The program continues and displays each entry in the table, as specified in message EZY1095I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2095I **Mt Elem=counter, ModeName=name, Flag=flag, Model=model, Class=class, DeviceType=type**

Explanation: Information for specific VTAM Telnet mode table entries. All table entries including number, the mode name, table name, and any flags are displayed.

System Action: The program continues and displays each entry in the table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2096I **LU Group Table**

Explanation: Information notifying you that TCPIP is about to display the logical unit group (LUGROUPS) table used by VTAM. This table contains an entry for each group defined with the LUGROUP statement. Each entry contains the group name, and a set of LU names that belong to the group.

System Action: The program continues, and displays the VTAM LU names groups definition table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2097I Lg Ptr=*table*, ElemCount=*counter*, Size=*bytes*

Explanation: Information specifying the allocated storage, pointed to in the *table* value, for the VTAM logical unit group (LUGROUPS) table. The *counter* value specifies the number of elements in the mode table. The *bytes* value specifies the size of the entries in the mode table.

System Action: The program continues and displays each entry in the table, as specified in message EZY2098I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2098I Lg Elem=*element*, GrpName=*groupID*, AllocCnt=*count*, UsedCnt=*count*

Explanation: Information about a specific element in the VTAM logical unit group (LUGROUPS) table. The *element* value specifies the element number of the LU group table. The *groupID* value specifies the element number of the LU group table. The *count* values specifies the dynamic storage allocated and the storage used up by the LU group table.

System Action: The program continues and displays each element in the LU group table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2099I Lg Mem=*LUmember*, MemberName=*LUname*

Explanation: Information about a specific VTAM logical unit group (LUGROUPS) table. The *LUmember* value specifies the member of the LU group table. The *LUname* specifies the name for the member of the LU group table.

System Action: The system continues, and displays all present members of the LU group table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2100I IP Group Table

Explanation: Information notifying you that TCPIP is about to display the Internet Protocol (IP) group table used by VTAM. This table contains an entry for each IP address and Subnet addresses defined with the IPGROUP statement. Each entry contains either an IP address or a subnet address/mask pair and the group name the address belongs to.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2101I **Ig** *Ptr=table, ElemCount=counter, Size=bytes, SubnetIndex=index*

Explanation: Information specifying the allocated storage, pointed to in the *table* value, for a new VTAM internet protocol group (IPGROUP) table. The *counter* value specifies the number of elements in the IP group table. The *bytes* value specifies the size of the entries in the IP group table. The *index* value specifies the start of IP subnets in the IP group table.

System Action: The program continues and displays each entry in the IP group table, as specified in messages EZY2102I, EZY2103I, and EZY2104I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2102I **Ig** *Elem=counter, GrpName=name,*

Explanation: Information indicating the IP group name where the VTAM table entries reside.

System Action: The system continues and displays message EZY2103I with additional information related to this message.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2103I **Ig** *MemberAddr=network.subnet.host.local, IsSubnet=flag*

Explanation: The IP address information for a specific member of the IP group table entries is displayed. The *flag* indicates if the subnet mask address is either *ON*, in use, or *OFF*, not used. If the subnet mask address is used it will be displayed in message EZY2104I following this message.

System Action: The system continues displaying message EZY2104I if the subnet address is in use. Otherwise, the system continues displaying IP group member addresses.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2104I **Ig** *Ip SubnetMask=network.subnet.host.local*

Explanation: This message appears only if the subnet mask is in use. The IP address information for the subnet mask of a IP group table member is displayed. The subnet mask address is in dotted-decimal notation. It is used as a mask, or a filter for the member address with which it is associated. The Subnet allows specification of IP address ranges and all the hosts on a sub-network. The Subnet is specified with a Subnet address and Subnet mask. The Subnet address is ANDed with the Subnet mask to remove any insignificant bits and stored as an IP address. The Subnet mask must be ANDed with the Telnet IP address and compared with the above Subnet address to determine if the IP address belongs to the subnet. The mask is stored after the Subnet address as an IP address.

System Action: The system continues displaying IP group member addresses.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2105I IP to LU Map Table

Explanation: Information notifying you that TCPIP is about to display the map internet protocol (IP) address groups to VTAM logical unit (LU) names table. This table contains an entry for each mapping defined with the LUMAP statement. Each entry contains either an IP address or an IP group name that is mapped to a VTAM LU name or an LU group name.

System Action: The program continues and displays the information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2106I Mp Ptr=*table*, ElemCount=*counter*, Size=*bytes*, AddrIndex=*index*

Explanation: Information specifying the allocated storage, pointed to in the *table* value, for a new VTAM internet protocol group (IP) to logical unit (LU) map table. The *counter* value specifies the number of elements in the IP to LU map table. The *bytes* value specifies the size of the IP to LU map table. The *index* value specifies the index to address a new part in the IP to LU map table.

System Action: The program continues and displays each entry in the IP to LU map table, as specified in messages EZY2107I, EZY2108I, and EZY2109I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2107I Mp Elem=*counter*, LU id=*id*, IsGroup=*groupflag*, IsHost=*hostflag*

Explanation: Information for specific LU ID of the IP to LU map table. The table element number, and the LU identification number, *id*, are displayed. The elements can be a VTAM LU name or an LU group name. The *groupflag* is a flag used to identify if this LU element is a VTAM LU name or an LU group. The *hostflag* is a flag used to identify if this LU element has a host IP address or a group name.

System Action: The system continues displaying message EZY2108I if the this element is a host, or the system displays message EZY2109I to display the IP group name.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2108I Mp Ip Address=*network.subnet.host.local*

Explanation: This information is only displayed if the LU element is a host. The IP address information for a specific host of the IP to LU map table entries is displayed.

System Action: The system continues displaying the IP to LU map table elements.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2109I Mp Ip GroupName =*name*

Explanation: This information is only displayed if the LU element is an IP group name. The group name of the IP to LU map element is displayed.

System Action: The system continues displaying the IP to LU map table elements.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2110I Default Application Table

Explanation: Information notifying you that TCPIP is about to display the default application (appl) table used by VTAM.

This table contains an entry for each mapping defined with the DEFAULTAPPL or the LINEMODEAPPL statement. Each entry contains either an IP address, an IP group name, or a linkname that is mapped to an application name. This application becomes the default application when the Telnet user has not specified one.

The order of the search is IP address, IP group name, and finally the Linkname. There is also a special match-all entry that is used as the default if Telnet session has no default application explicitly defined.

System Action: The system continues displaying the default application table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2111I Da Ptr=*table*, ElemCount=*counter*, Size=*bytes*, AddrIndex=*index*

Explanation: Information specifying the allocated storage, pointed to in the *table* value, for a new VTAM default application (appl) table. The *counter* value specifies the number of elements in the default appl table. The *bytes* value specifies the size of the default appl table. The *index* value specifies the default appl table index to the IP addresses part.

System Action: The program continues and displays each entry in the default appl table, as specified in messages EZY2112I, EZY2113I, and EZY2114I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2112I Da Elem=*counter*, ApplName=*name*, LogMode=*mode*, IsHost=*hostflag*

Explanation: Information for specific entries of the default application (appl) table. The table element number, the VTAM appl name, as specified in VTAMLST, and the logon *mode*, are displayed. The logon mode *mode* can be either line or transparent mode. The *hostflag* variable is a flag used to identify if this default element has a host IP address.

System Action: The system continues displaying message EZY2113I if this default entry has the IsHost variable set to ON. The true condition identifies this element as being a host, any other condition and the system displays the destination name in message EZY2109I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2113I Da Ip Address=*network.subnet.host.local*

Explanation: This information is only displayed if the default element is a host. The IP address information for a specific host of the default table entries is displayed.

System Action: The system continues displaying the default table elements.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2114I Da Ip GroupName =*name*

Explanation: This information is only displayed if the default element is either an IP group name or a linkname. The group name of the default element is displayed.

System Action: The system continues displaying the default table elements.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2115I Application Table

Explanation: Information notifying you that TCPIP is about to display the application (appl) table used by VTAM.

This table contains the applications which are permitted access through Telnet as specified by the RESTRICTAPPL or the ALLOWAPPL statement. The application entries contain an optional RestrictUsers table identifying the RESTRICTed applications, or an AllowLuNames set identifying the ALLOWed applications.

System Action: The system continues displaying the application table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2116I Ap Ptr=*table*, **ElemCount**=*counter*, **Size**=*bytes*

Explanation: Information specifying the allocated storage, pointed to in the *table* value, for a new VTAM application (appl) table. The *counter* value specifies the number of elements in the application table. The *bytes* value specifies the size of the appl table.

System Action: The program continues and displays each entry in the appl table, as specified in the following messages:

- EZY2117I
- EZY2118I
- EZY2119I
- EZY2120I
- EZY2121I
- EZY2122I

User or Operator Response: None.

System Programmer Response: None.

EZY2117I • EZY2120I

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2117I **Ap Elem=counter, ApplName=name, ApplMask=X'mask'**

Explanation: Information for specific entry of the application (appl) table. The table element number, the VTAM appl name, as specified in VTAMLST, and the appl comparison mask *mask*, are displayed. The appl mask can be used for logical comparisons to validate characters in the appl name and user IDs specified on the RESTRICTAPPL and the ALLOWAPPL statements.

System Action: The system continues displaying message EZY2118I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2118I **Ap AlocCnt=count, UsedCnt=count, Discon=Discflag, Restrict=Rstrflag**

Explanation: Information about a specific element in the VTAM application (appl) table. The *count* values specifies the dynamic storage allocated and the storage used up by the appl table. The *Discflag* allows an application to be disconnected by a user leaving the VTAM session active. The *Rstrflag* identifies this application as a restricted appl if it is set to the ON flag.

System Action: The program continues and displays each element in the appl table. If the Rstrflag is set ON, then the program continues and displays the restricted applications in messages EZY2119I, EZY2120I, and EZY2121I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2119I **Ap Elem=counter, UserName=id, UserMask=X'mask'**

Explanation: This information is only displayed if the *Rstrflag* is set to ON, in message EZY2118I, specifying a restricted table member. This information is for specific restricted users of the application (appl) table. The table element number, the user name ID, as specified in RESTRICTAPPL statement, and the user comparison mask *mask*, are displayed. The user ID mask can be used for logical comparisons to validate characters in the user IDs specified on the RESTRICTAPPL statements.

System Action: The system continues displaying message EZY2120I and EZY2121I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2120I **Ap AlocCnt=count, UsedCnt=count,**

Explanation: This information is only displayed if the *Rstrflag* is set to ON, in message EZY2118I, specifying a restricted table member. This information is about a specific element in the VTAM application (appl) table. The *count* values specifies the dynamic storage allocated and the storage used up by the appl table.

System Action: The program continues and displays each element in the appl table, and continues to message EZY2121I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2121I **Ap Elem=counter, RestrictLuName=name**

Explanation: This information is only displayed if the *Rstrflag* is set to *ON*, in message EZY2118I, specifying a restricted table member. This information is about a specific element in the VTAM application (appl) table. The *counter* values specifies the appl table entry. The *name* specifies the restricted user LU name for that entry in the appl table.

System Action: The program continues and displays each restricted user LU names in the appl table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2122I **Ap Elem=counter, AllowLuName=name**

Explanation: This information is about a specific element in the VTAM application (appl) table. The *counter* value specifies the appl table entry. The *name* specifies the allowed user logical unit (LU) name for that entry in the appl table. This table contains the applications which are permitted access through Telnet as specified by the ALLOWAPPL statement.

System Action: The program continues and displays each allowed user LU names in the appl table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2130I **LU Status Table**

Explanation: Information notifying you that TCPIP is about to display the logical unit (LU) status table used by VTAM. The table is used to keep track of the status of VTAM LU names as they are used in Telnet connections.

System Action: The system continues and displays messages EZY2131I, and EZY2132I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2131I **Ls Ptr=number, AllocCnt=count, UsedCnt=count**

Explanation: Information specifying the allocated and used logical units pointed to in the logical unit (LU) status table used by VTAM.

System Action: The system continues and displays message EZY2132I.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2132I Ls Elem=*number*, Name=*luname*, Flags=*Ls_Default flag*

Explanation: Displays specific information about the entries in the logical unit (LU) status table. The LU element numbers *number*, names *luname*, and any optional flags *flgs* are displayed.

System Action: The system continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPBLVT

Procedure Name: Mainline code

EZY2133E VTAM UssTable *table name* can not be loaded.

Explanation: This message displays if the debug option is used. The VTAM unformatted systems service table indicated could not be loaded.

System Action: TCPIP continues.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Make sure that the table name exists and that the program has access to it. Call the IBM Software Support Center for more help if necessary.

Source Data Set: MVPDG7D

Procedure Name: MVPDG7DD

EZY2134I STOP COMMAND IGNORED tcipprocname ALREADY BEING STOPPED

Explanation: This message indicates that a subsequent stop command (P TCPIP) was received while TCPIP was processing a previous stop command. The subsequent stop command is ignored and TCPIP ends.

System Action: TCPIP ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPOCM

Procedure Name: MVPOCMI

EZY2142I Hung VMCF Interrupt scheduled for ASID

Explanation: This message indicates that TCPIP has encountered a VMCF interrupt pending and the interrupt processing for the client is not running.

System Action: TCPIP continues. The VMCF interface is hung.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Reinitiate the VMCF interface and reissue the interrupt if required. If the problem continues, contact the IBM Software Support Center.

Source Data Set: MVPXVI

Procedure Name: MVPXVI

EZY2143I Hung IUCV_A Interrupt scheduled for ASID

Explanation: This message indicates that TCPIP has encountered a IUCV_A interrupt pending and the interrupt processing for the client is not running.

System Action: TCPIP continues. The IUCV application is hung.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Restart the IUCV application and reissue the interrupt if required. If the problem continues, contact the IBM Software Support Center.

Source Data Set: MVPXVI

Procedure Name: MVPXVI

EZY2144I Hung IUCV_C interrupt scheduled for ASID

Explanation: This message indicates that TCPIP has encountered a IUCV_C interrupt pending and the interrupt processing for the client is not running.

System Action: TCPIP continues. The IUCV interface is hung.

User or Operator Response: Refer this message to the system programmer.

System Programmer Response: Reinitiate the IUCV interface and reissue the interrupt if required. If the problem continues, contact the IBM Software Support Center.

Source Data Set: MVPXVI

Procedure Name: MVPXVI

EZY2145I VMCF cleanup for USERID *id* in ASID

Explanation: This message indicates that VMCF is running cleanup for the indicated user ID.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

Procedure Name: MVPXVI

EZY2370I *ssock_max number*

Explanation: This message displays the maximum number of simultaneous sockets available for use. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: getdtablesize

EZY2371I Initialized echo server

Explanation: The echo server is initialized and ready to process incoming UDP requests.

System Action: The application continues normal processing.

User or Operator Response: To remove informational messages, start the echo server without any trace options. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for information about the echo server.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_socket

EZY2372I Initialized TCP echo server

Explanation: The echo server is initialized and ready to process incoming TCP requests.

System Action: The application continues normal processing.

User or Operator Response: To remove informational messages, start the echo server without any trace options. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the echo server.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_TCPsocket

EZY2373I Initialized discard server

Explanation: The discard server is initialized and ready to process incoming UDP requests.

System Action: The application continues normal processing.

User or Operator Response: To remove informational messages, start the discard server without any trace options. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the discard server.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_socket

EZY2374I Initialized TCP discard server

Explanation: The discard server is initialized and ready to process incoming TCP requests.

System Action: The application continues normal processing.

User or Operator Response: To remove informational messages, start the discard server without any trace options. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the discard server.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_TCPsocket

EZY2375I Initialized character generator server

Explanation: The character generator server is initialized and ready to process incoming UDP requests.

System Action: The application continues normal processing.

User or Operator Response: To remove informational messages, start the character generator server without any trace options. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the character generator server.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_socket

EZY2376I Initialized TCP character generator server

Explanation: The character generator server is initialized and ready to process incoming TCP requests.

System Action: The application continues normal processing.

User or Operator Response: To remove informational messages, start the character generator server without any

trace options. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the character generator server.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_TCPsocket

EZY2380I before selectex: readmask

Explanation: This message displays the contents of the readmask used by the function selectex to determine if a socket is prepared to read data. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: main

EZY2381I exiting

Explanation: This message is displayed to indicate that the miscellaneous server is closing all sockets and exiting the application.

System Action: The miscellaneous server halts.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: main

EZY2385I after selectex: readmask

Explanation: This message displays the value of the readmask used by the function selectex to determine if a socket is ready to read data.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: main

EZY2386I socket *number* port *number* is condition

Explanation: This message indicates that data has been received over the indicated socket. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: main

EZY2387I before cextpost

EZY2388I ended interrupt handler: *rc value*

Explanation: The interrupt handler is being closed. The *rc* portion of this message indicates whether the closing is successful. This message is displayed when the debug trace option is specified.

System Action: The interrupt handler halts. The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: cextpost

EZY2389I UDP echo protocol

Explanation: A UDP echo request is being processed. The server echoes the data received. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: doecho

EZY2390I Echo server: socket *number* address *name*

Explanation: This message displays the socket value and the address family (in dotted decimal notation) for the echo server. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about AF_INET or AF_IUCV.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: doecho

EZY2391I UDP discard protocol

Explanation: The UDP discard protocol is initialized. The UDP discard protocol will discard the data received by the MISCSRV.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dodisc

EZY2392I Discard server: socket *number* address *name*

Explanation: This message displays the socket value and address (in dotted decimal notation) for the discard server.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dodisc

EZY2393I UDP character generator

Explanation: The UDP character generator is initialized. This will cause random data to return for each datagram that is received. This message is displayed when the debug trace option is specified. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the character generator.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dochargen

EZY2394I Character server: socket *number* address *name*

Explanation: This message displays the socket number and address (in dotted decimal notation) for the character server.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISSCRV

Procedure Name: dochargen

EZY2395I link *name name* family *name* home address *name*

Explanation: This message states the link name, family and home address of all the system adapter interfaces.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: getifconf

EZY2396I *name* socket *number* port *number*

Explanation: The link name, socket number, and well-known port number are displayed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_socket

EZY2397I **socket** *number* **port** *number*

Explanation: The socket number and well-known server port number are displayed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: init_TCPsocket

EZY2398I **socket** *number* **af** *number*

Explanation: The socket number and address family number, which indicates an AF_IUCV connection, are displayed for the SMSG socket.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: SMSGsocket

EZY2399I **TCP echo protocol**

Explanation: A TCP echo request is being processed. The server echoes the data received. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: doTCPecho

EZY2401I **TCP Discard protocol**

Explanation: A TCP discard request is being processed. The server discards the data received. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: doTCPdisc

EZY2403I **TCP Character Generator protocol**

Explanation: A TCP character generator request is being processed. The character generator server creates random data in response to data received. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: doTCPchargen

EZY2405I Display SMSG

Explanation: An SMSG command is being processed. Shutdown and trace commands are the only supported SMSG commands for the miscellaneous servers. See the *z/OS Communications Server: IP Configuration Guide* and the *z/OS Communications Server: IP User's Guide and Commands* for more information about the SMSG command. This message is displayed when the debug trace option is specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dispSMSG

EZY2406I Received message:

Explanation: This message header is followed by information pertaining to the status of the SMSG message requested.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dispSMSG

EZY2414I shutting down

Explanation: The MISCSRV server is closing due to user or operator request. The server closes all connections and halts. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the SHUTDOWN function.

System Action: The application halts.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dispSMSG

EZY2416I Unknown command: *name*

Explanation: The MISCSRV server received a command that it does not recognize. The command is ignored.

System Action: The application continues.

User or Operator Response: Reissue the command using the SHUTDOWN command or TRACE command. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the SHUTDOWN or TRACE commands.

System Programmer Response: Assist the user if necessary.

Source Data Set: MISCSRV

Procedure Name: dispSMSG

EZY2418I **closed STREAM Socket:** *number rc: number*

Explanation: The miscellaneous server closes the socket specified in the message.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: closesock

EZY2419E **error for select** (*tcperror*)

Explanation: The select function was unsuccessful.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application attempts to continue.

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

System Programmer Response: Respond as indicated (*tcperror*). See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about select().

Source Data Set: MISCSRV

Procedure Name: selectex

EZY2420E **selectex** (*tcperror*)

Explanation: The selectex function was unsuccessful.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application attempts to continue.

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

System Programmer Response: Respond as indicated in(*tcperror*). See *selectex()* in the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information.

Source Data Set: MISCSRV

Procedure Name: selectex

EZY2421E **recvfrom()** (*tcperror*)

Explanation: The recvfrom function found no data while in nonblocking mode.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in (*tcperror*). For more information about the (*tcperror*) portion of this message, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: doecho

EZY2422E **sendto()** (*tcperror*)

Explanation: The buffer space that is used to hold the data to be transmitted is inadequate, or no data was found in the buffer space. If the socket is in nonblocking mode a return code EWOULDBLOCK is set.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. Refer to the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information on the `sendto()` function.

Source Data Set: MISCSRV

Procedure Name: `doecho`

EZY2424E `create_TCP_port::socket` (*tcperror*)

Explanation: An error occurred when trying to create a socket connection.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The MISCSRV halts. The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. For more information on the error condition see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: `getifconf`

Procedure Name: `init_TCPsocket`

EZY2425E `create_UDPC_port::socket` (*tcperror*)

Explanation: The function `create_UDPC_port`, which is used to open a UDP connection, was unsuccessful.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The UDP application MISCSRV halts.

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

System Programmer Response: Respond as indicated in *(tcperror)*. For more information on the error condition see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: `init_socket`

EZY2426E `create_TCP_port::bind` (*tcperror*)

Explanation: The `bind()` call encountered an error when trying to bind an address to the socket.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. See *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for a description of error values.

Source Data Set: MISCSRV

Procedure Name: `init_TCPsocket`

EZY2427E `create_UDPC_port::bind` (*tcperror*)

Explanation: The `bind()` function encountered an error when trying to bind a UDP port.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. For more information about the `bind()` function

EZY2428E • EZY2431E

and addition error values see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: init_socket

EZY2428E **create_TCP_port::listen** (*tcperror*)

Explanation: The listen function, which is used to create a TCP connection to listen for data, encountered an error. (*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in (*tcperror*). For more information about the listen function and additional error values see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: init_TCPsocket

EZY2429E **create_IUCV_port::socket** (*tcperror*)

Explanation: An error occurred when creating an IUCV_port and socket connection.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in (*tcperror*). For more information on the error condition see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: init_SMSGsocket

EZY2430E **cannot connect msg socket** (*tcperror*)

Explanation: An error was detected when trying to initiate a socket connection to the server. No connection is established.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in (*tcperror*). A description of the connect() function and error codes for the function can be found in the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: init_SMSGsocket

EZY2431E **Accept()** (*tcperror*)

Explanation: The accept() call, which is used to establish a connection to a client, failed to respond to a connection request.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. A description of the *accept()* call and the error conditions generated from this call can be found in the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: doTCPecho

EZY2432E *recv()* (tcperror)

Explanation: The *recv()* function, which is used to receive data from a socket and store it in a buffer, was unsuccessful.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The MISCSRV halts. The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. For more information about the *recv()* call and the error conditions for this call, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: doTCPecho

EZY2433E *error reading msg_socket* (tcperror)

Explanation: The buffer space used to store data read from the *msg_socket* cannot accommodate the data. The following errors can occur as a result of the failure:

EBADF The socket is not a valid socket descriptor.

EFAULT Exceeded storage for the caller address space.

EWOULDBLOCK

 The socket is in nonblocking mode and data is not available to read.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*. For more information see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: MISCSRV

Procedure Name: dispMSG

EZY2434E *create_UDPC_port: ioctl (get interface conf.)* (tcperror)

Explanation: The *ioctl* function, which is used to get the network configuration, encountered an error. For more information about the *ioctl* function, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

(tcperror) is the number of the message where you can find more detailed information about this error.

System Action: The MISCSRV server halts. The application continues.

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

System Programmer Response: Respond as indicated in *(tcperror)*.

Source Data Set: MISCSRV

Procedure Name: getifconf

EZY2435I **Echo server:** *socket number buf value*

Explanation: This message displays the server type, socket number, and contents of the buffer that holding data to be transmitted.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: doecho

EZY2436I **Discard server:** *socket number buf value*

Explanation: This message displays the server type, socket number, and contents of the buffer holding received data.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dodisc

EZY2437I **Character server:** *socket number buf value*

Explanation: This message displays the server type, socket number, and the data stored in the buffer.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dochargen

EZY2438I **Invalid Trace option** *name*

Explanation: The trace option requested is not valid. The following are valid trace options:

(NO)ECHO Returns the data for UDP and TCP sessions exactly as it is received.

(NO)DISC The data is discarded when sent to the MISCRSV.

(NO)CHARGEN Random data is displayed regardless of the data it receives.

System Action: The application continues.

User or Operator Response: Reenter the trace option with a valid request.

System Programmer Response: Assist the user as necessary.

Source Data Set: MISCSRV

Procedure Name: dispMSG

EZY2439I **Currently not tracing anything.**

Explanation: The trace option is not enabled.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dispMSG

EZY2440I **Currently tracing:** *name*

Explanation: This informational message displays the current level of tracing.

System Action: The application continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: dispMSG

EZY2441I **Trace option not defined**

Explanation: The application does not support the trace option.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MISCSRV

Procedure Name: main

EZY2442I **Invalid initialization parameter:** *name*

Explanation: The parameter you have entered is not valid.

System Action: The application continues.

User or Operator Response: Restart the miscellaneous server with a valid parameter. See the *z/OS Communications Server: IP User's Guide and Commands* for more information about correct parameters.

System Programmer Response: Assist the user as necessary.

Source Data Set: MISCSRV

Procedure Name: main

EZY2443I **fcntl failed:** (*tcperror*)

Explanation: The fcntl() call failed to initialize the socket descriptor into nonblocking mode.

(*tcperror*) is the number of the message where you can find more detailed information about this error.

System Action: Initialization of the MISCSRV server halts. The application continues.

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

System Programmer Response: Respond as indicated in (*tcperror*). See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for more information about the fcntl() call and the error conditions that can occur when using this call.

Source Data Set: MISCSRV

Procedure Name: init_TCPsocket

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

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

EZY2633E • EZY2635E

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

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

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

Source Data Set: EZAFTPEP

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

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

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

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

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

Source Data Set: EZAFTPEP

EZY2634E *parameter value value not numeric.*

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

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

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

System Programmer Response: Correct the FTP.DATA file or the FTP server procedure 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 and FTP server start options.

Source Data Set: EZAFTPEP

EZY2635E *Integer overflow number for parameter*

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

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

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

System Programmer Response: Correct the FTP.DATA file or FTP server start procedure 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 and FTP server start options.

Source Data Set: EZAFTPEP

EZY2636E *parameter value not specified.*

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

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

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

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

Source Data Set: EZAFTPEP

EZY2638I **Using FTP configuration defaults.**

Explanation: The FTP server was unable to locate/open any external FTP.DATA files. The internal defaults for all FTP.DATA file parameters are being used. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

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

User or Operator Response: None.

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

Source Data Set: EZAFTPEP

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

Explanation: This message indicates which FTP.DATA file is being used for the configurable server parameters. *name* will either be "DD:SYSFTPD", indicating that the SYSFTP DD statement was used for the FTP.DATA file, or it will be the actual name of the file being used. See the *z/OS Communications Server: IP Configuration Reference* for information about the FTP.DATA file search order.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPEP

EZY2642E **Unknown keyword: *parameter***

Explanation: While processing the FTP.DATA file, the FTP server encountered a parameter which it did not recognize.

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

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

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

Source Data Set: EZAFTPEP

EZY2643E *parameter value longer than number characters.*

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

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

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

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

Source Data Set: EZAFTPEP

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

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

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

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

EZY2645E **CONDDISP value must be either CATLG or DELETE.**

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

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

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

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

Source Data Set: EZAFTPEP

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

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

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

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

System Programmer Response: Correct the value of the FILETYPE parameter in the FTP.DATA file to be either SEQ, JES, or SQL. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZY2651E UNITNAME parameter unit: invalid unit type.

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

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

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

System Programmer Response: Correct the value of the UNITNAME parameter in the FTP.DATA file to be a valid DASD or TAPE unit. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZY2655E Error in start parameters.

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

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

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

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

Source Data Set: EZAFTPEP

EZY2657W Host name unknown; gethostname() error: error

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

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

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

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

Source Data Set: EZAFTPDM

EZY2658W domain name unknown: gethostbyname() error.

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

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

User or Operator Response: Contact the system programmer.

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 Reference* for information on domain name systems or HOSTS.SITEINFO and HOSTS.ADDRINFO data sets.

Source Data Set: EZAFTPDM

EZY2659E RECFM value must be one of: F, FA, FB, FBA, FBM, FBS, FBSA, FBSM, FM, FS, FSA, FSM, V, VA, VB, VBA, VBM, VBS, VBSA, VBSM, VM, VS, VSA, VSM, U, UA, or UM.

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

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

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

System Programmer Response: Correct the value of the RECFM parameter in the FTP.DATA file to be a valid record format. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

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

System Programmer Response: Correct the value of the SPACETYPE parameter in the FTP.DATA file to be a valid space type. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

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

System Programmer Response: Correct the value of the SQLCOL parameter in the FTP.DATA file to be a valid column heading. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZY2671I General tracing is already active.

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

EZY2672E read_ftpdata invalid keycode encountered: *key_code*

Explanation: While processing the FTP.DATA file, an internal error occurred.

key_code is an internal code describing the error.

System Action: An assertion failure occurs to provide additional information about the error.

User or Operator Response: Contact the system programmer with the error message and the assertion failure output.

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

Source Data Set: EZAFTPEP

EZY2673E Error reading FTP configuration file: *error_description*

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

error_description describes the error.

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

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

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

Source Data Set: EZAFTPEP

EZY2674E LRECL number is invalid for RECFM *recfm*.

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

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

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

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM or LRECL. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZY2675I LRECL being reset to *number*

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

System Action: LRECL parameter is changed.

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

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

Source Data Set: EZAFTPEP

EZY2676E BLOCKSIZE must equal LRECL for RECFM *recfm*.

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

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

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

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM, LRECL, or BLOCKSIZE or all three. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZY2677I BLOCKSIZE being set to *blocksize number*.

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

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

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

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

Source Data Set: EZAFTPEP

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

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

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

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

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM, LRECL, or BLOCKSIZE or all three. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

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

System Programmer Response: Update the FTP.DATA file with the corrected values for RECFM, LRECL, or BLOCKSIZE or all three. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZY2680I LRECL being changed to *number*

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

System Action: LRECL is changed to the indicated value.

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

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

Source Data Set: EZAFTPEP

EZY2681E Invalid record format 'recfm' encountered.

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

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

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

System Programmer Response: Correct the FTP.DATA statement by choosing the valid values.

Source Data Set: EZAFTPEP

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

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

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

User or Operator Response: Contact the system programmer if you do not want to use the default values.

System Programmer Response: Correct the FTP.DATA statement by choosing the valid values.

Source Data Set: EZAFTPEP

EZY2688E Unknown start option: *option*

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

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

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

System Programmer Response: Correct the start option in the FTP server start procedure. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the FTP Server start options.

Source Data Set: EZAFTPEP

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

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

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

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

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

Source Data Set: EZAFTPNY

EZY2690E Header record invalid format.

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

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

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

EZY2691E • EZY2700I

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

Source Data Set: EZAFTPNY

EZY2691E Error reading the file: *error*

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

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

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

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

Source Data Set: EZAFTPNY

EZY2693I Unable to open *name* : *reason*

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

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

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

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

Source Data Set: EZAFTPCZ, EZAFTPEP, EZAFTPNX

EZY2697I IBM FTP *version time on date*

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

EZY2700I Using port FTP control (*portnumber*)

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

EZY2701I Inactivity time is *timeout*

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDS

EZY2706E Unsupported Modify command parameter *parameter* ignored.

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

System Action: The request is ignored.

User or Operator Response: Verify that the Modify command was entered correctly. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the Modify command.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

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

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

System Action: The MVS command is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

EZY2714I FTP server shutdown in progress

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

System Action: The FTP daemon terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDPH

EZY2715I General tracing is not active.

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPHD

EZY2720I Using Japanese translation tables in *filename*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

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

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

EZY2722E Unable to load Japanese translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

EZY2723I Using Korean translation tables in *filename*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

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

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

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

System Action: FTP continues.

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

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

System Programmer Response: None.

Source Data Set: EZAFTPDC

EZY2725E Unable to load Korean translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

EZY2726I Using traditional Chinese translation tables in *filename*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

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

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

EZY2728E Unable to load traditional Chinese translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPRU

EZY2792E Error retrieving "data_set". Dcbdsn parameter ignored.

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

System Action: FTP continues.

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

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPRU

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPRU

EZY2794E Error mounting "data_set". Dcbdsn parameter ignored.

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

System Action: FTP continues.

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

System Programmer Response: Assist the user as necessary.

Source Data Set: EZAFTPRU

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPRU

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPRU

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

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

System Action: FTP continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Change the DCBDSN parameter of the FTP.DATA file to specify a data set that is on a direct access volume. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

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

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

System Action: FTP continues.

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

System Programmer Response: Change the DCBDSN keyword in the FTP.DATA file to specify a non-VSAM data set. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPRU

EZY2799E "data_set" invalid dsorg. Dcbdsn parameter ignored.

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

System Action: FTP continues.

EZY2800E • EZY2832E

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

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

Source Data Set: EZAFTPURU

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

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

System Action: FTP continues.

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

System Programmer Response: Change the DCBDSN keyword in the FTP.DATA file to specify a valid data set. Refer to the *z/OS Communications Server: IP Configuration Reference* for information on the parameters of the FTP.DATA file.

Source Data Set: EZAFTPURU

EZY2830I Using simplified Chinese translation tables in *filename*

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

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDC

EZY2831E Unable to load simplified Chinese translation tables in *filename*; return code: *code*

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

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

EZY2832E Unable to load simplified Chinese translation tables.

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

System Action: FTP continues.

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

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

Source Data Set: EZAFTPDC

Chapter 4. EZY3xxxx messages

EZY3720I *f=number d=number rl=hexvalue rdl=number pdh=number pdl=number*

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: print_request_parms

EZY3721I *rc=number err=number rpl=hexvalue rpb=hexvalue rpbl=number*

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: print_request_parms

EZY3722I *string hexvalue xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx*

Explanation: Hexadecimal display of information starting at storage address *hexvalue*. This display is generated by a call to the prntdata function. The *string* value is supplied by the caller to identify this output. If more than 16 bytes of data are displayed, message EZY3723I is used for additional lines.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: prntdata

EZY3723I *string xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx*

Explanation: Hexadecimal storage display generated by a call to the prntdata function. This format is used for subsequent lines after message EZY3722I until the requested amount of data is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: prntdata

EZY3724I **descarray is at *addr*, size is *bytes bytes***

Explanation: This indicates the address and size of the socket descriptor array.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: alloc_global_fdsets

EZY3725I **descarray has *number* entries, entry size is *bytes***

Explanation: This indicates the number of socket descriptor entries and the size of the entries that appear in the IUCV socket descriptor table.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: alloc_global_fdsets

EZY3726I **iucvdesc is at *addr***

Explanation: The IUCV descriptor structure is located at the address in this message. This is the location of extra storage for IUCV socket information.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: alloc_global_fdsets

EZY3727I **in nextbuf with *function* type**

Explanation: The function descarray_nextbuf has been called with a type argument other than receive, send, read, accept, or write. The function descarray_nextbuf issues this message and returns -1.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_nextbuf

EZY3728I **data pending=***data*, **fd=***descriptor*, **path=***path*, **iptype=***type* (*status*)

Explanation: This indicates the pointers to the number of input buffers left to be read, the file descriptor, the IUCV path ID, the IUCV parameter list status type, and the actual status.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_iucv_receive

EZY3729I **fd=***descriptor* **iucvselect now TRUE**

Explanation: This indicates the file descriptor and that at least one AF_IUCV descriptor is in the caller's mask.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_select

EZY3730I **fd=***descriptor* **inetselect now TRUE**

Explanation: This indicates the file descriptor and that at least one AF_INET descriptor is in the caller's mask.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_select

EZY3731I **IUCV interrupt from TCPIP,**

Explanation: This message is issued when debugging is requested (a trace socket statement appears in the tcpip.data configuration file). It is issued along with message EZY3732X to indicate that an IUCV interrupt has been received from TCPIP.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3732X *fd=descriptor, path=path, type=type (string)*

Explanation: This message is concatenated to EZY3731I and indicates the file descriptor, the path ID, the numeric value of the interrupt type, and the symbolic type of external interrupt.

System Action: TCPIP continues

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3733E *userid severed IUCV path.*

Explanation: The indicated user severed the IUCV path. The TCP connection corresponding to each stream socket is reset.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3734E *userid severed IUCV path. Reason .ipuser.field*

Explanation: The indicated userid severed the IUCV path. This occurs only in the case of shutdown or an unexpected error. The ipuser field in the SEVER external interrupt indicates the reason for the sever. The reason is in EBCDIC. The possible reason codes and explanations appear in *z/OS Communications Server: IP Programmer's Guide and Reference*.

System Action: The system action depends on the type of error that occurred.

User or Operator Response: None.

System Programmer Response: See *z/OS Communications Server: IP Programmer's Guide and Reference* to determine the reason for this error and to correct the error. If the error is the result of an error in TCPIP, contact the IBM Software Support Center.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3735I **IUCV interrupt, fd=descriptor, path=path, type=type (string)**

Explanation: This message is issued when socket tracing is requested and indicates an IUCV interrupt of the indicated type number (name) occurred for the file descriptor and path ID specified.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3736S **Had a rejected message completion, msg lost**

Explanation: An IUCV interrupt denoting "rejected message" was received; the message is lost.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3737W Sever Pending Connection from *addrspace* for *application*

Explanation: An IUCV interrupt was received for a socket that does not exit. The connection to the indicated TCPIP address space is severed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3738W Sever Pending Connection, no user data

Explanation: The pending connection of the IUCV path to the TCPIP address space has been severed. No user data is available for the connection.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: descarray_save_iucvbuf

EZY3739W Message Complete audit field: *value.value.value*

Explanation: An IUCV interrupt occurred with nonzero audit data. This signifies an error was detected after the IUCV operation was requested. The values in the message indicate more specific details about the type of error.

System Action: Error return will be indicated by the socket function invoked.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the traces and the dump.

Source Data Set: CMIUCSOC

Procedure Name: check_audit

EZY3740I Rc=*code* on IUCV *function* fd=*descriptor*, buf (ipbfadr2) is at *address*

Explanation: This message appears during a request to get the next IUCV buffer if socket tracing is in effect. The return code from the IUCV_NEXTBUFFER function, the IUCV function, the file descriptor, and the address of the buffer for the request are displayed.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

EZY3741I • EZY3744E

Procedure Name: doiucv

EZY3741I **Rc=code on IUCV_function to id fd=descriptor, path=path, iprcode=iprcode, ipmsgid=hexid, iucvname=string**

Explanation: This message displays information about an IUCV function including:

- Return code
- Function name
- User ID of the TCPIP address space
- File descriptor
- Path ID
- The isolated pacing response (IPR) code
- The message ID from the function
- IUCV name or descriptor name

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: doiucv

EZY3742I **ciucv_data area (ipbfadr2) is at address**

Explanation: The IPBFADR2 data area is at the location pointed to by the address in the message.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: doiucv

EZY3743I **Caught a REPLY after PURGE**

Explanation: A reply external interrupt occurred after the IUCV_PURGE function. This message is generated only when socket tracing is in effect. IUCV ignores the reply.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: doiucv

EZY3744E **Rc=code on IUCV_function to id, path=path, iprcode=code, iucvname=name**

Explanation: This message displays information about an unsuccessful IUCV function including:

- Return code
- Function name
- User ID of the TCPIP address space
- Path ID
- The isolated pacing response (IPR) code

- IUCV name or descriptor name

This message appears when socket tracing is not in effect (when socket tracing is specified, other messages convey the same information as this one, therefore this message is not generated.)

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: CMIUCSOC

Procedure Name: doiucv

EZY3745E **Should not be in sock_request_iucv for function,**

Explanation: The sock_request_iucv routine should not have been called for the function indicated in this message. IUCV returns a -1 return code and an error to indicate that an incorrect file number was detected.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: This is an internal error. Contact the IBM Software Support Center to report the error.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_iucv

EZY3746I **sock_request_inet entry parms:**

Explanation: This message precedes the list of parameters that are being used for the sock_request_inet routine, the routine that sends an INET socket request.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3747I **sock_request_inet returns EIBMIUCVERR**

Explanation: The sock_request_inet routine encountered an error while trying to send an INET socket request. This message has additional text concatenated to it that gives more information about the circumstances where the error occurred.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3748X after check_and_establishiucv() err

Explanation: This message is concatenated to message EZY3747I. The current IUCV path has been severed. The socket application attempted to establish a new path, but was not successful. The application sets the error number parameter to indicate an IUCV error, and the return code to indicate no path ID.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3750X after doiucv() err

Explanation: This message is concatenated to message EZY3747X. An error occurred when IUCV sent an INET socket request to TCPIP.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3752X due to bad irpt

Explanation: This message is concatenated to message EZY3747X. An error occurred when an IUCV interrupt was received. Information in the interrupt buffer does not match what was expected.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3754X after check_audit()

Explanation: This message is concatenated to message EZY3747I. The check_audit routine checked the audit field of an IUCV interrupt and the field indicated an error occurred during the interrupt. The return code and errno value provide more information about the error. IUCV returns the return code and errno value in the parameter list.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3755I sock_request_inet return parms:

Explanation: These are the parameters that the sock_request_inet routine returns upon completion. If an error occurred during this routine, the return code and errno will appear in this parameter list.

This message is part of the socket trace used for problem analysis and can be generated by specifying **trace socket** in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: sock_request_inet

EZY3756I flush_dgram: about to call sock_request(), totlenbufferlength

Explanation: The sock_request function is to be called, with a buffer of the length indicated in the message. This call will transmit all locally buffered datagrams to TCPIP. This message is generated only when a socket is operating in bulk mode. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_send

EZY3757I sending iovcnt=number

Explanation: This is the number of datagrams being sent for a socket request. Message EZY3758I will be generated for each of these datagrams. This message is generated only when a socket is operating in bulk mode. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_send

EZY3758I theiovs [number].iov_base= number.iov_len=length data=

Explanation: The iovec structure fields (address and length of a datagram) is displayed, followed by the first 50 bytes of the datagram in hexadecimal format. This message is generated only when a socket is operating in bulk mode. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_send

EZY3774I *fd=descriptor* in callers rmask

Explanation: This message identifies a file descriptor set in the caller's read mask. The message will be issued once for each file descriptor that is set in the read mask. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3775I *fd=descriptor* unallocated

Explanation: The indicated file descriptor is not allocated, but it was specified in a read, write, or exception mask supplied by an application as an argument to a select function. This message is generated only when socket tracing is in effect. The select function returns an error to indicate an incorrect file number.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3776I *fd=descriptor* in callers wmask

Explanation: This message identifies the file descriptors set in the caller's write mask. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3777I *fd=descriptor* in callers emask

Explanation: This message identifies the file descriptors set in the caller's exception mask. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3778I Select would cause deadlock

Explanation: The arguments to select specified nothing (no file descriptor, no time-out, etc.) that could be or become ready, therefore an attempt to perform this select function would never return. Error code EDEADLK is returned to caller.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3779I in inetselect

Explanation: The status of the AF_INET descriptors in the inetselect mask is being checked (an interaction with TCPIP will occur). This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3780I in inetselect iucverror

Explanation: The socket IUCV path to the TCPIP address space no longer exists. (This might occur if TCPIP is unsuccessful, or is intentionally stopped, for example.) This message is generated only when socket tracing is in effect. Error code -1 is generated by the select function called by the application.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3781I waitvalue of 0 secs to TCPIP

Explanation: A select function has already determined a selected AF_IUCV descriptor is ready. It is forcing a wait time of zero when contacting TCPIP to learn whether any selected AF_INET descriptor is also ready. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3782I in iucvselect, iucvnfds=*numberfds*

Explanation: This message indicates the number of ready AF_IUCV file descriptors found in the select mask. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3783I user specified *seconds s micro-seconds ms* wait

Explanation: The user requested the specified time-out value in a call to a select function. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3784I issue stimer now

Explanation: A select function was called with a time-out argument, but no AF_INET descriptors in the selection mask and no other selected descriptors ready. Therefore, the select function will use STIMER to measure the wait time locally. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3785I do not issue stimer

Explanation: A select function was called with a time-out value of zero. No selected descriptor is ready, and no wait will be performed. The select function sets the user file descriptor masks to zero (no descriptor is ready) and returns a zero return code. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3786I wait forever, null ptr to timeval

Explanation: A select function has been called with NULL as the time-out argument. No selected descriptor is currently ready, therefore the socket application will wait until a descriptor becomes ready (or the specified ECB is posted, if selectex was called). This message is generated only when socket tracing is in effect.

System Action: The application waits until the desired action occurs.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3787I wait ecblist=*list*, ecbcoun=*count*

Explanation: Displays the address of the ECB list, and the number of ECBs in this list, that will be used by a select function to wait for an event to occur. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3788I **iucvposted=*iucv*, waitposted=*wait*, callposted=*call***

Explanation: Describes the result of the WAIT operation described by message EZY3787I, specifically indicating the posted status of the ECB used for IUCV interrupts (iucvposted), the ECB used for timer interrupts (waitposted), and the ECB supplied by the calling application (callposted). This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3789I **in iucvposted**

Explanation: A select function is processing an IUCV interrupt. If the interrupt indicates a message complete from TCPIP, IUCV merges the AF_IUCV file descriptor sets with the AF_INET file descriptor sets returned by TCPIP. If there are AF_IUCV descriptors in the caller's masks, any IUCV interrupt selected by the caller's mask completes the request and the caller's mask is set accordingly. If an AF_IUCV interrupt is received for a descriptor not in the caller's mask, the select function continues waiting. This message is generated only when socket tracing is in effect.

System Action: The application's action depends on the specific case. See the explanation for more information. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3790I **iucvcomp is now TRUE**

Explanation: A select function found at least one AF_IUCV descriptor in the caller's masks, and an IUCV interrupt occurred on an AF_IUCV path. Assignment of TRUE to the variable iucvcomp causes later code to check whether this IUCV interrupt satisfies the select condition. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3791I **in gotmsgcomp**

Explanation: This indicates that select received a Message Complete interrupt from TCPIP (because one of the AF_INET descriptors identified by the select function's caller is ready) and is about to proceed accordingly. Select adds any AF_IUCV descriptors that are ready to the AF_INET ready descriptors in the mask, and continues. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

EZY3792I • EZY3795I

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3792I *nfds=value, return=value*

Explanation: Displays the number of AF_IUCV descriptors that are ready (nfd) and the total number of ready descriptors (return), which is the value returned by select. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3793I *must wait for TCPIP*

Explanation: This indicates that a select function must wait for a reply to its message to TCPIP before it can complete. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3794I *in iucvcom && iucvselect*

Explanation: A select function is ending because an AF_IUCV descriptor has become ready before any AF_INET descriptor or other action that would end the select (for example, time-out or ECB posting). This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3795I *in wait posted*

Explanation: A select function is ending because the time-out value specified by the caller has expired. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3796I in wait posted iucvselect

Explanation: A select function's time-out period has expired (this message follows message EZY3795I) and the selection mask includes one or more AF_IUCV descriptors. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3797I callers ECB posted

Explanation: A select function is ending because the ECB identified by the caller has been posted, rather than any of the specified descriptors becoming ready. Select sets the caller's masks to zero and returns a zero return code.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: common_select

EZY3818I Your *dataset* has enabled debug.

Explanation: Debugging has been requested by an option in your TCPIP.DATA data set. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: ReadProf

EZY3819I Your *dataset* has enabled TESTSTOR.

Explanation: The TESTSTOR function has been requested by an option in your TCPIP.DATA data set. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: ReadProf

EZY3820I Your *dataset* has disabled TESTSTOR.

Explanation: The TESTSTOR function has been turned off by an option in your TCPIP.DATA data set. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

EZY3821I • EZY3829I

Source Data Set: CMIUCSOC

Procedure Name: ReadProf

EZY3821I TESTSTOR is enabled.

Explanation: The TESTSTOR function is active and will check the validity of storage addresses provided as arguments in calls to socket functions. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: ReadProf

EZY3822I TESTSTOR is not enabled.

Explanation: The TESTSTOR function is not active. This message is generated only when socket tracing is in effect.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CMIUCSOC

Procedure Name: ReadProf

EZY3828E *dataset* could not be opened or is not defined.

Explanation: The ETC.SERVICES data set could not be opened or is not defined.

System Action: GETSERVE continues, additional services are ignored.

User or Operator Response: None.

System Programmer Response: If the application ends abnormally as a result of requiring this *hlq*.ETC.SERVICES data set, create the data set implicitly using a valid high level qualifier or explicitly using the DD:SERVICES definition in the application's catalogued procedure. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: GETSERVE

Procedure Name: setservent

EZY3829I pre xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request, getibmopt

EZY3830I xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request,getibmopt

EZY3831I xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request,getibmopt

EZY3832I xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request,getibmopt

EZY3833I xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request,getibmopt

EZY3834I xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

EZY3835I • EZY3952X

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request, getibmopt

EZY3835I post xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

Explanation: This message is part of the socket trace used for problem analysis and can be generated by specifying trace socket in the TCPIP.DATA configuration file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Software Support Center and, if requested to do so, provide this information.

Source Data Set: CMIUCSOC

Procedure Name: hpns_common_select, hpns_do_init, hpns_do_term, hpns_sock_request, getibmopt

EZY3950X No error

Explanation: This message indicates normal completion, no Inter-User Communication Vehicle (IUCV) error is found. The socket call is successful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3951X Invalid Path-id

Explanation: The path ID in an IUCV operation is incorrect. The socket call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3952X Path is quiescent

Explanation: The IUCV path used in a send operation has been quiesced by the receiver. No messages can be sent on this path until the receiver has performed a resume operation.

System Action: The socket call continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events

leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3953X Msg limit Exceeded

Explanation: The maximum number of IUCV messages has already been queued for the receiver on the path over which a new message send request has been made. The IUCV message is discarded. An error return is generated by the socket function the application called.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3954X Priority msg not allowed on this path

Explanation: The IUCV CONNECT parameter for the connection listed priority = NO, so no priority messages are allowed on this path. The message is discarded.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3955X Reply too big

Explanation: An IUCV SEND operation specified a reply was required, but the length of the reply sent by the message recipient exceeded the size of the answer buffer provided by the message sender for this reply.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3956X Fetch Protection Exception on answer buffer

Explanation: The address specified for the reply message by an IUCV REPLY operation points to an address in storage that is fetch protected.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3957X Addressing Exception on answer buffer

Explanation: The pointer to the reply buffer specified in an IUCV operation is incorrect or null.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3958X Conflicting msg class/path/msgid

Explanation: This message indicates one of three conditions:

- an unrecognized socket call constant was found in the high-order halfword of the target message class (TRGCLS) keyword.
- conflicting path ID
- conflicting message ID

IUCV operation is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3959X Msg was purged

Explanation: An IUCV REPLY was performed for a message that has been purged by the sender. The reply message is discarded.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3960X Negative msg length

Explanation: The buffer length (BUFLN) or data length (PRMMSG) is negative and is therefore an incorrect parameter for an IUCV operation. The IUCV call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3961X Target userid not logged on

Explanation: The target communicator specified for an IUCV CONNECT function does not exist. The IUCV call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3962X Target userid not enabled for IUCV

Explanation: The target communicator specified for an IUCV CONNECT function has not enabled IUCV operations. The IUCV call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3963X Exceeded number of paths (this side)

Explanation: You have exceeded the maximum of 255 simultaneous connections allowed under IUCV. The connection is not established.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3964X Exceeded number of paths (other side)

Explanation: The target communicator has exceeded the maximum of 255 simultaneous connections allowed under IUCV. The IUCV call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3965X Not authorized

Explanation: You are not authorized to establish an IUCV connection with the remote user's address space. The IUCV call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3967X Invalid function code

Explanation: Unknown function specified for IUCV operation. No operation is performed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software

Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3968X Msg limit exceeds 255

Explanation: The message limit specified for an IUCV CONNECT operation is too large. The IUCV CONECT operation is ignored.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3969X Buffer already declared

Explanation: An IUCV buffer had already been declared, the first step in initializing for IUCV operations, when another IUCV DECLARE BUFFER request was made.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3970X Path was severed by other side

Explanation: The IUCV partner has severed the path for which an IUCV ACCEPT function is requested. The ACCEPT function is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3971X Cannot accept data in parmlist

Explanation: An IUCV SEND operation specifying data to be delivered in the parameter list could not be performed because the CONNECT operation that established the IUCV path to be used for the SEND operation did not specify that data could be delivered in the parameter list. The IUCV operation is unsuccessful.

EZY3972X • EZY3974X

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3972X SEND buffer list is invalid

Explanation: The send buffer list data is incorrect or null. The IUCV operation is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3973X Negative length in answer list

Explanation: The length specified in the IUCV socket call for the buffer list (BUFLEN) is negative. The socket call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3974X Total list length is invalid

Explanation: A buffer list was specified for an IUCV SEND or REPLY operation, and the total message length specified does not match the sum of the individual lengths of the message pieces in the buffer list. The IUCV call ends.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3975X PRMSG and BUF/ANSLIST conflict

Explanation: The PRMSG option is not allowed when you specify ANSLIST = YES. The message is not sent.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3976X Buffer list not on double word boundary

Explanation: The buffer list for an IUCV call must begin on an appropriate boundary in storage. The IUCV call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3977X Answer list not on double word boundary

Explanation: The answer list for an IUCV call must begin on an appropriate boundary in storage. Like the BUFLIST field (see message EZY3976X), the ANSLIST must begin on a double-word boundary. The IUCV call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3978X Must create control buffer

Explanation: An attempt was made to perform an IUCV operation, such as CONNECT, on a control path but no DECLARE BUFFER has been issued for control paths. All IUCV operations default to application path.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of

EZY3985E • EZY3988E

socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3985E Bad socket-call constant (EIBMBADCALL)

Explanation: An incorrect socket-call constant was found in the IUCV header, such as a code that refers to a non-existent socket function. The socket call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the arguments in the socket call. For details, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3986E Bad parm (EIBMBADPARM)

Explanation: One of the parameters specified in the socket call is incorrect. The socket call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the offending parameter descriptor. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3987E Socket out of range (EIBMSOCKOUTOFRANGE)

Explanation: The socket descriptor is outside the range accepted for the IBM implementation of sockets. The socket call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the socket descriptor for the IBM socket to an acceptable value. For details, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3988E Socket in use (EIBMSOCKINUSE)

Explanation: The socket number assigned by the client application for accept, socket, or takesocket call is already in use. The socket call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Create a new accept, socket, or takesocket call for the application. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3989E IUCV error (EIBMIUCVERR)

Explanation: The request failed due to an IUCV error. This error is generated by the client stub code. The socket call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the IUCV error message that appears directly after this message to aid in determining the nature of the error. For more information about IUCV sockets, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3993E Conflict with other call on socket (EIBMCONFLICT)

Explanation: A previous function on the same socket has not completed and the present call is not of a type that can be processed or queued until the previous operation completes. The present call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3994E Call canceled (EIBMCANCELLED)

Explanation: A close() call has been issued by either the server or the client.

System Action: If the socket is closed when input data is queued, the TCP connection is reset rather than being cleanly closed.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3996E A TCP/IP name that is not valid was detected (EIBMBADTCPNAME)

Explanation: A TCP/IP name that was not valid was detected. The socket call is unsuccessful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: The system derives the TCP/IP value from the configuration file, as described in the *z/OS Communications Server: IP Configuration Reference*. Correct the TCPIPJOBNAME value defined in the TCPIP.DATA file that is allocated to the job.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY3997E TSRB request code is invalid (EIBMBADREQUESTCODE)

Explanation: A request code that was not valid was detected.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Correct the arguments in the socket call. For details, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. After the error occurs, obtain a dump of the VMCF address space. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

Chapter 5. EZY4xxxx messages

EZY4000E A connection token that is not valid was detected. (EIBMBADCONNECTIONMATCH)

Explanation: A connection token that is not valid was detected. There is no such connection.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4001E TCPIP Abend (EIBMTCPABEND)

Explanation: An abend occurred in TCP processing of this request.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4008E TCPIP C Socket Abend (EIBMTERMERROR)

Explanation: TCP/IP C Socket encountered a terminating error.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4011E The delete requestor did not create the connection. (EIBMINVDELETE)

Explanation: The request was called from an invalid task.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4012E The connection token validation failed. (EIBMINVSOCKET)

Explanation: Connection token validation failed. The socket does not exist.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4013E Connection Terminated by TCPIP (EIBMINVTCPCONNECTION)

Explanation: A socket connection was terminated by TCP/IP.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4021E TCPIP not installed or not active (EIBMNOACTIVETCP)

Explanation: TCPIP is not installed or not active.

System Action: TCPIP not available.

User or Operator Response: Issue a D TCPIP command to determine whether TCPIP is installed. If it is installed, restart TCPIP and reissue the socket call.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4022E **The requested control block contained data that is not valid. (EIBMINVTSRBUSERDATA)**

Explanation: The requested control block contained data that is not valid.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message, the traces, and the dump.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4026E **Client received bad post code (EIBMBADPOSTCODE)**

Explanation: This message is received by an outstanding (blocked) socket when TCPIP is stopped. It indicates that an application was given a bad post code.

System Action: TCPIP ends.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces will provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4050E **No error**

Explanation: This message indicates normal completion of a socket call.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the application is correct in detecting that some error occurred, another program has reset the value of errno before the call to tcperror generated this message. Otherwise, the application logic has incorrectly decided an error occurred, and is calling tcperror to display a message when no reason exists to do so.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4051E **Not owner (EPERM)**

Explanation: Permission is denied. The caller does not have correct ownership to perform the requested operation.

System Action: The call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Check the permissions for the data set or directory.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4052E No such file or directory (ENOENT)

Explanation: A referenced data set or directory does not exist.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Verify the data set or directory name, and correct the reference.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4053E No such process (ESRCH)

Explanation: The referenced process or task was not found.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Verify the process name, and correct the reference.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4054E Interrupted system call (EINTR)

Explanation: A function was unable to complete, usually because of conditions having nothing to do with the request itself. This code does not diagnose an error in a request, but reports that the system handling the request found it more convenient to abandon this request than to continue it after some other activity was performed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Repeat the original request.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4055E I/O error (EIO)

Explanation: Some sort of I/O error occurred while the operation was in progress. This could be a disk error, for example.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Make sure that sufficient storage exists for newly-created data sets. Use the appropriate I/O support functions under MVS to isolate the source of the error.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4056E No such device or address (ENXIO)

Explanation: The process name or driver name for the socket call is not found.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Verify the local and remote addresses and process names for the socket call. For more information about socket calls, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4057E Arg list too long (E2BIG)

Explanation: The argument list for the call is too long.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the arguments to the call. Reinitiate the application.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4058E Exec format error (ENOEXEC)

Explanation: The sequence of calls in the client or server program is incorrect. The socket call cannot be executed as written.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Check the sequence of calls within the application program. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4059E Bad file number (EBADF)

Explanation: An invalid argument to the socket call was specified. The nature of the error depends on the particular call that triggers it.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters of the appropriate socket call. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for information about System Error Return Codes.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4060E No children (ECHILD)

Explanation: No daughter processes exist. Generated by some X-Window System functions.

System Action: TCPIP continues.

User or Operator Response: None.

EZY4061E • EZY4064E

System Programmer Response:

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4061E No more processes (EAGAIN)

Explanation: There are no more processes queued to this connection.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4062E Not enough memory (ENOMEM)

Explanation: A memory allocation (malloc) call for an operation failed.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters of the appropriate socket call. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for information about System Error Return Codes.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4063E Permission denied (EACCES)

Explanation: The caller does not have the correct permissions to perform the requested operation.

System Action: The request fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters of the appropriate socket call. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for information about System Error Return Codes.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4064E Bad address (EFAULT)

Explanation: An incorrect storage address or length was specified in the socket call. The following list displays some of the possible function calls that may return this error value:

Call	Explanation
accept	Using ADDR and ADDRLLEN would result in an attempt to copy the address into a portion of the caller's address space into which information cannot be written.
bind	Using NAME and NAMELEN would result in an attempt to copy the address into a nonwritable portion of the caller's address space.

Call	Explanation
connect	Using NAME and NAMELEN would result in an attempt to copy the address into a portion of the caller's address space into which data cannot be written.
send	Using the MSG and LEN parameters would result in an attempt to access storage outside the caller's address space.
takesocket	Using the CLIENTID parameter as specified would result in an attempt to access storage outside the caller's address space.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this to your application and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4065E Block device required (ENOTBLK)

Explanation: An inappropriate kind of I/O device was specified for an operation.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response:

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4066E Device busy (EBUSY)

Explanation: This message indicates that listen () has already been called for the socket.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct or delete the givesocket() call. Retry the task.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4067E File exists (EEXIST)

Explanation: The specified data set already exists. Your socket call would overwrite it.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the write() call.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

EZY4068E • EZY4071E

Procedure Name: tcperror

EZY4068E Cross-device link (EXDEV)

Explanation: A socket call specified a cross-device link.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Respecify the parameters of the appropriate socket call to prevent the cross-device link.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4069E No such device (ENODEV)

Explanation: The device pointed to in the givesocket() does not exist.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the client ID parameter in the givsocket() call.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4070E Not a directory (ENOTDIR)

Explanation: The caller specified a non-directory in a directory operation.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the pointer in the socket call parameter.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4071E Is a directory (EISDIR)

Explanation: The caller specified a directory in a non-directory operation.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the pointer in the socket call parameter.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4072E Invalid argument (EINVAL)

Explanation: The argument specified to the socket call was not valid. The nature of the error depends on the particular call that triggers it. The following list displays some of the possible function calls that may return this error value:

accept	Listen was not called for this socket.
bind	The socket is already bound to an address.
connect	The specified name length is incorrect.
fcntl	Incorrect flags were specified.
givesocket	An incorrect client ID was entered.
ioctl	The request is incorrect or not supported.
select	One of the fields in the time-out structure is incorrect.
sendto	The target address length is incorrect for the specified address family.
shutdown	The shutdown condition is not 0, 1, or 2.
takesocket	The specified client ID is incorrect.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the above table as a guide for the possible conditions under which this error value can occur. Apply this to your application and, correct the parameters of the appropriate socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4073E File table overflow (ENFILE)

Explanation: The data set containing the connection table for the TCPIP address space exceeded its maximum size.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the `maxdesc()` call to increase the maximum number of sockets from the default value of 47. Then use the `getdtablesize()` call to verify the new maximum. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4074E Too many open files (EMFILE)

Explanation: The socket descriptor table is already full. The following list displays some of the possible function calls that may return this error value:

maxdesc	Indicates that <code>*totdesc</code> is greater than 2000.
takesocket	The socket descriptor table is already full.

System Action: The call stops. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

EZY4075E • EZY4078E

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4075E Inappropriate ioctl for device (ENOTTY)

Explanation: An incorrect device call was specified in the cmd parameter of the ioctl() call.

System Action: The socket call closes. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change or delete the ioctl() command for the device. For more information about the ioctl() call, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4076E Text file busy (ETXTBSY)

Explanation: The text data set pointed to by the call is currently being written to by another process.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the socket call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4077E File too large (EFBIG)

Explanation: The data set is too large. The operation caused the data set to grow beyond the server's limit.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the protocol or socket type specified in the socket() call. For more information about the socket() call, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4078E No space left on device (ENOSPC)

Explanation: Insufficient storage exists on the target device to carry out the operation. The operation caused the server's file system to reach its limit.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the len parameter of the send() call. For more information about the send() call, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4079E Illegal seek (ESPIPE)

Explanation: The stream pipe specified in the socket call is not correct.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the read() call parameters. For more information about the read() call, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4080E Read-only file system (EROFS)

Explanation: Your application cannot write to the data set specified in the socket call. The data set resides on a read-only file system.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4081E Too many links (EMLINK)

Explanation: The host connection queue is already full.

System Action: The call is bypassed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4082E Broken pipe (EPIPE)

Explanation: One of the stream pipes used to redirect I/O in the socket call has failed.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Perform a SOCKET or IUCV trace to determine where the failure occurred. For more information, refer to *z/OS Communications Server: IP Diagnosis Guide*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4083E Argument too large (EDOM)

Explanation: The parameter for the socket call was too large to be a valid argument to the call.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the socket call parameter. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4084E Result too large (ERANGE)

Explanation: The value returned by the socket call is out of the range of anticipated values.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the socket call parameters as necessary and retry the call.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4085E Operation would block (EWOULDBLOCK)

Explanation: The meaning and severity of this message depend on the call that generates it. The following list displays some of the possible function calls that may return this error value:

accept	The socket is in nonblocking mode and connections are not queued. This is not an error condition.
---------------	---

read recvfrom	The socket is in nonblocking mode and read data is not available. This is not an error condition.
----------------------	---

send sendto write	The socket is in nonblocking mode and buffers are not available.
--------------------------	--

System Action: The socket call is bypassed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4086E Operation now in progress (EINPROGRESS)

Explanation: The socket is marked nonblocking and the connection cannot be completed immediately. The following list displays some of the possible function calls that may return this error value:

connect	The socket descriptor <i>s</i> is marked nonblocking, and the connection cannot be completed immediately. The EINPROGRESS value does not indicate an error condition.
----------------	---

System Action: The socket call is bypassed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this to your application and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4087E Operation already in progress (EALREADY)

Explanation: The operation specified in the socket call is already in progress. The following list displays some of the possible function calls that may return this error value:

connect	The socket descriptor <i>s</i> is marked nonblocking, and a previous connection attempt has not completed.
----------------	--

maxdesc Your program called maxdesc() after creating a socket, or after a previous call to maxdesc().

System Action: The socket call is closed. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4088E Socket operation on non-socket (ENOTSOCK)

Explanation: A socket operation was requested on a non-socket. The socket was incorrectly defined.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the socket call parameters.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4089E Destination address required (EDESTADDRREQ)

Explanation: A destination address is needed for the socket call to complete.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Add the destination address to the socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4090E Message too long (MSGSIZE)

Explanation: The message length specified in the sendto call is too large. The maximum is 32,767. The following list displays some of the possible function calls that may return this error value:

sendmsg The message was too big to be sent as a single datagram. The default is large-envelope-size.

sendto The message was too big to be sent as a single datagram. The default is large-envelope-size.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4091E Protocol wrong type for socket (EPROTOTYPE)

Explanation: The specified protocol is incorrect for this socket type.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the protocol parameter for the socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4092E Option not supported by protocol (ENOPROTOOPT)

Explanation: The socket option specified is incorrect for the protocol type or the level is not SOL_SOCKET. The following list displays some of the possible function calls that may return this error value:

getsockopt The OPTNAME parameter is unrecognized, or the LEVEL parameter is not SOL_SOCKET.

setsockopt The OPTNAME parameter is unrecognized, or the LEVEL parameter is not SOL_SOCKET.

getibmssockopt The OPTNAME parameter is unrecognized, or the LEVEL parameter is not SOL_SOCKET.

setibmssockopt The OPTNAME parameter is unrecognized, or the LEVEL parameter is not SOL_SOCKET.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4093E Protocol not supported (EPROTONOSUPPORT)

Explanation: The protocol specified in the socket call is not supported by the address domain or the socket type. The following list displays some of the possible function calls that may return this error value:

socket The PROTOCOL is not supported in this domain, or this PROTOCOL is not supported for this socket type.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4094E Socket type not supported (ESOCKTNOSUPPORT)

Explanation: The socket type specified in the socket call is not supported.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the type parameter in the socket() call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4095E Operation not supported on socket (EOPNOTSUPP)

Explanation: The selected socket is not a stream socket. The following list displays some of the possible function calls that may return this error value:

accept The s parameter is not of type SOCK_STREAM.
givesocket The socket type is not SOCK_STREAM.
listen The s parameter is not a socket descriptor that supports the listen() call.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4096E Protocol family not supported (EPFNOSUPPORT)

Explanation: The protocol specified in the socket call is not supported by the address family specified. The following list displays some of the possible function calls that may return this error value:

getclientid The domain is not AF_INET.
takesocket The domain field of the CLIENTID parameter is not AF_INET.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4097E Address family not supported by protocol family (EAFNOSUPPORT)

Explanation: The address family specified in the socket call is not supported by the protocol specified. It must be either AF_IUCV or AF_INET. The protocol field must be set to 0 if the domain parameter is set to AF_IUCV. The following table displays some of the possible function calls that may return this error value:

bind The address family is not supported (it is not AF_IUCV or AF_INET).
connect The address family is not supported.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

EZY4098E • EZY4100E

Procedure Name: tcperror

EZY4098E Address already in use (EADDRINUSE)

Explanation: The bind() call cannot bind the address specified to the socket with descriptor s because the address is already in use. The following table displays some of the possible function calls that might return this error value:

bind	The address is already in use. See the SO_REUSEADDR option described under getsockopt() and the SO_REUSEADDR option described under setsockopt() in the <i>z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference</i> for more information. This error will also occur if the port specified on the bind has been configured as RESERVED in a TCP/IP profile port reservation statement.
-------------	--

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4099E Can't assign requested address (EADDRNOTAVAIL)

Explanation: Either the address specified in the socket call is incorrect for the bind() call or the calling host cannot reach the address specified in the connect() call. The following list displays some of the possible function calls that may return this error value:

bind	The address specified is not valid on this host. For example, the internet address does not specify a valid network interface.
-------------	--

connect	The calling host cannot reach the specified destination.
----------------	--

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters for the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4100E Network is down (ENETDOWN)

Explanation: The host network is down.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the application when the host network restarts.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4101E Network is unreachable (ENETUNREACH)

Explanation: The network of the address specified in the name parameter of the connect() call cannot be reached from this host. The following list displays some of the possible function calls that may return this error value:

bind The network cannot be reached from this host.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4102E Network dropped connection on reset (ENETRESET)

Explanation: The connection was dropped when the socket was reset.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the socket call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4103E Software caused connection abort (ECONNABORTED)

Explanation: The application program caused the connection to abend.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use sock_debug() to determine why the connection abended. For more information about tracing and debugging socket calls, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4104E Connection reset by peer (ECONNRESET)

Explanation: If a stream socket is closed by the remote process while there is input data queued, the TCP connection is reset rather than being cleanly closed.

System Action: The connection is reset.

User or Operator Response: None.

System Programmer Response: The socket call is suspended.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4105E No buffer space available (ENOBUFS)

Explanation: The specific meaning of this message depends on the call that triggers it. The following list displays some of the possible function calls that may return this error value:

all	No buffer space is available.
accept	Not enough buffer space is available to create the new socket.
send sendto write	Not enough buffer space is available to send the new message.
takesocket	There is a socket control block (SCB) or socket interface control block (SKCB) shortage in the TCPIP address space.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4106E Socket is already connected (EISCONN)

Explanation: The socket specified by the socket descriptor, *s*, in the connect() call is already in use. The following list displays some of the possible function calls that may return this error value:

Call	Explanation
connect	The socket descriptor <i>s</i> is already connected.

System Action: The socket call continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4107E Socket is not connected (ENOTCONN)

Explanation: The socket specified by the descriptor, *d*, in the givesocket() call is not connected. The following list displays some of the possible function calls that may return this error value:

getpeername	The socket is not in the connected state.
givesocket	The socket is not connected.

System Action: The call is unsuccessful.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4108E Can't send after socket shutdown (ESHUTDOWN)

Explanation: A send() call cannot be carried out after a shutdown() call.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Rewrite the socket call so that the shutdown() follows the send(). The only call that can follow the shutdown() is the close() for the socket.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4109E Too many references: can't splice (ETOOMANYREFS)

Explanation: A socket call had too many references. The splice could not be completed.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Rewrite the socket call to reduce the number of references.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4110E Connection timed out (ETIMEDOUT)

Explanation: The connection time to live expired before the socket call was completed. The following list displays some of the possible function calls that may return this error value:

connect The connection establishment timed out before a connection was made.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4111E Connection refused (ECONNREFUSED)

Explanation: The connection request was rejected by the remote server. The following table displays some of the possible function calls that may return this error value:

connect The connection request was rejected by the destination host.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the table above as a guide to the conditions under which this error can occur. Apply this information to your application, and correct the parameters of the appropriate socket call. For more information, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4112E Too many levels of symbolic links (ELOOP)

Explanation: A pointer for a socket call parameter has too many levels of symbolic links for TCP to handle the call.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Redefine the pointer for the socket call parameter in such a way that it has fewer levels of symbolic links.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4113E File name too long (ENAMETOOLONG)

Explanation: The data set name pointed to by the socket call is too long to be handled by TCPIP. The maximum number of characters is 248.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the pointer for the socket call parameter so that the data set pointed to has a shorter name.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4114E Host is down (EHOSTDOWN)

Explanation: The host machine is down.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the call after the host restarts.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4115E Host is unreachable (EHOSTUNREACH)

Explanation: A connection to the host machine cannot be established.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Ensure that there is a network path to the specified host and that the host name is valid. If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. The traces will provide more information about the internal operations of socket functions and might suggest a reason for the error, or yield some better insight into the chronology of events leading up to the error. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4116E Directory not empty (ENOTEMPTY)

Explanation: The caller attempted to remove a directory that was not empty.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4117E Too many processes (EPROCLIM)

Explanation: Too many processes are running on the system.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Rewrite the socket call so as to close some of the active processes. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4118E Too many users (EUSERS)

Explanation: Too many users are on the system.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the socket call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4119E Disc quota exceeded (EDQUOT)

Explanation: The client's disk quota on the server has been exceeded.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the len parameter of the write() call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4120E Stale NFS file handle (ESTALE)

Explanation: The Network File System (NFS) is unable to mount the MVS data set because the file handle that points to the data set no longer exists, or access to it has been revoked. Under NFS, each data set is referenced by the client through a file handle, which is a 32-bit identifier field.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the MOUNT command to update the data set handle. For more information about NFS, see *MVS/DFP Version 3 Release 3 Customizing and Operating the Network File System Server*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4121E Too many levels of remote in path (EREMOTE)

Explanation: The address structure pointed to by the socket call has too many levels of remote qualifiers.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Modify the address or name parameter for the socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4122E Not a stream device (ENOSTR)

Explanation: A stream socket type was declared in the socket() call, but the remote device cannot transfer stream data.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the socket type to SOCK_DGRAM in the socket() call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4123E Timer expired (ETIME)

Explanation: The timer for the socket call expired. The maximum interval to wait for completion was exceeded.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: If this message recurs, modify the timeout parameter of the select() call or change it to a NULL pointer. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4124E Out of stream resources (ENOSR)

Explanation: No stream sockets are available to service the socket call.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the client application.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4125E No message of desired type (ENOMSG)

Explanation: There is no message of the correct data type for the socket.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4126E Not a data message (EBADMSG)

Explanation: The message is a system or network control message.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4127E Identifier removed (EIDRM)

Explanation: The access identifier for the file or data set has been removed.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the mkpriv command to define the user privilege for the data set.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4128E Deadlock situation detected/avoided (EDEADLK)

Explanation: A deadlock, the condition that occurs when multiple processes are waiting for the availability of a resource that will not become available because it is held by another process that is in a similar wait state, was detected.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. Review the trace output to determine why the deadlock occurred. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4129E No record locks available (ENOLCK)

Explanation: The host system does not provide for the locking of data sets.

System Action: The socket call continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPEERROR

Procedure Name: tcperror

EZY4130E Machine is not on the network (ENONET)

Explanation: The machine specified in the socket call is not on the host network.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

EZY4131E • EZY4134E

System Programmer Response: Verify the address information in the socket call and modify as necessary.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4131E Object is remote (ERREMOTE)

Explanation: The data set or disk specified is not on the local network.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the address pointer in the socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4132E The link has been severed (ENOLINK)

Explanation: The physical connection linking the sockets has been broken at some point.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Retry the socket call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4133E Advertise error (EADV)

Explanation: The path specified in the adv (advertise) or share command is incorrect, and the data set, file, or disk cannot be accessed.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Verify the correct resource name and pathname. Reinitiate the socket call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4134E srmount error (ESRMNT)

Explanation: An error occurred when the Network File System (NFS) server attempted to place the remote file system in the user's data set hierarchy.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the mount command (# mount) to mount the remote resource. When you mount a remote resource, it stays mounted only during the current session.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4135E Communication error on send (ECOMM)

Explanation: A communication error has occurred on a send() call.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: If the error can be reproduced, set **trace socket** to active in the TCPIP.DATA configuration file and rerun the application. Review the trace output to determine why the error occurred. If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4136E Protocol error (EPROTO)

Explanation: A protocol error was detected for the socket call.

System Action: The socket call abends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4137E Multihop attempted (EMULTIHOP)

Explanation: The socket call specified a multihop address link. That is, it specified a connection path that omitted some routing information.

System Action: The socket call is bypassed.

User or Operator Response: None.

System Programmer Response: Correct the address parameters for the socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

If you are still unable to determine the cause of the problem, re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4138E Cross mount point (not an error) (EDOTDOT)

Explanation: The file system is mounted from a remote machine.

System Action: The socket call continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4139E Remote address changed (EREMCHG)

Explanation: The remote address specified in the socket call has changed.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Verify the remote address and correct the address parameters of the socket call.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4195E Arg list too long (E2BIG)

Explanation: The argument list for the call is too long.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the arguments to the call. Reinitiate the application.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4196E Invalid argument (EINVAL)

Explanation: The argument specified to the socket call was not valid. The nature of the error depends on the particular call that triggers it. The following list displays some of the possible function calls that may return this error value:

Call	Explanation
accept	Listen was not called for this socket.
bind	The socket is already bound to an address.
connect	The specified name length is incorrect.
fcntl	Incorrect flags were specified.
givesocket	An incorrect client ID was entered.
ioctl	The request is incorrect or not supported.
select	One of the fields in the time-out structure is incorrect.
sendto	The target address length is incorrect for the specified address family.
shutdown	The shutdown condition is not 0, 1, or 2.
takesocket	The specified client ID is incorrect.

System Action: The socket call ends. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the information provided in the above table as a guide for the possible conditions under which this error value can occur. Apply this to your application and, correct the parameters of the appropriate socket call. For more information, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4197E Bad file number (EBADF)

Explanation: An invalid argument to the socket call was specified. The nature of the error depends on the particular call that triggers it.

System Action: The socket call fails. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters of the appropriate socket call. See the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* for information about System Error Return Codes.

If you are still unable to determine the cause of the problem, re-create the problem with CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4198E Last IUCV call (IUCV call) returned error number

Explanation: This message displays the error number returned from the IUCV call. It is usually followed by a message in the range of EZY3950X–EZY3978X that explains the IUCV error.

System Action: The system action depends on the nature of the message that follows this one. For more information, see that specific message number.

User or Operator Response: None.

System Programmer Response: The system action depends on the nature of the message that follows this one. For more information, see that specific message number.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4199E Unknown errno value=error number

Explanation: The error number displayed is not recognized as one of the values used by TCPIP to report an error condition. For more information about system error codes, see *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Re-create the problem with the CTRACE option SOCKET active. Contact the IBM Software Support Center with the original message and the traces.

Source Data Set: TCPERROR

Procedure Name: tcperror

EZY4200E CallTransform ConvType Error : conversion_type

Explanation: The conversion type found does not match supported values (1–8). The supported values are:

Value	Conversion Type
1	SJISKANJI
2	JIS83KS
3	JIS78KJ
4	DECKANJI
5	EUCKANJI
6	KSC5601
7	HANGUEL
8	TCHINESE

System Action: Control is returned to the TNSTMAS routine, where the session is terminated.

EZY4201E • EZY4204E

User or Operator Response: Restart the session, specifying a valid conversion type.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: TnDbcsTm

EZY4201E CallTransform Verb Error. Verb : Verb

Explanation: The verb specified in the configuration data set does not match the supported options. The supported options are: TERMINIT, CHEKTYPE, ASKTTYTYPE, TTYHELLO, MAPTOASC, GRFTOMAP, MAPTOGRF, ASCTOMAP, BREAKKEY, TTLOGOFF, TTYGDBYE, SHUTDOWN, and SIMCHECK.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -4.

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

System Programmer Response: Correct the configuration data set to specify a proper verb from the list above. For more information on configuring the Telnet server, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TnDbcsTm

EZY4202E CALLTRANSFORM Terminal Id Error. Id : Term_Id

Explanation: The terminal ID was found to be outside expected range. It is either negative or greater than the maximum number of Telnet server connections. The return code is set to -5.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated.

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

System Programmer Response: Make sure that the terminal ID is properly specified. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TnDbcsTm

EZY4203E TNDBCSTM - Telnet DBCS Transform Mode - ERRLOG DDname not found - TNDBCSEr.

Explanation: The Telnet server was unable to allocate ddname TNDBCSEr. This is where all error messages for TELNET 3270 DBCS transform are written.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -1.

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

System Programmer Response: Update the JCL or TCPIP procedure to include TNDBCSEr.

Source Data Set: TNDTRSVB

Procedure Name: Simcheck

EZY4204E TNDBCSTM - Telnet DBCS Transform Mode - Config DDName not found - TNDBCSCN

Explanation: The Telnet server was unable to allocate ddname TNDBCSCN. This is the TELNET 3270 DBCS transform configuration data set.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -99.

User or Operator Response: None.

System Programmer Response: Update the JCL or TCPIP procedure to include TNDBCSCN.

Source Data Set: TNDTRSVB

Procedure Name: Simcheck

EZY4206I IBM TCP/IP TELNET SERVER DBCS SERVICE START AT *time* ON *date*

Explanation: This message indicates the time and date at which the Telnet session started.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: TTYHELLO

EZY4207E TTYHELLO CodeConversion error. rc : *rc*

Explanation: An error was encountered in the Code Conversion routine used to convert message EZY4206I during a call to the procedure TTYHELLO. The return code in this message indicates the offset in the current input buffer at which the error occurred. This message is written to the error log.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -2.

User or Operator Response: None.

System Programmer Response: Check the input data to determine why this error occurred.

Source Data Set: TNDTRSVB

Procedure Name: TTYHELLO

EZY4208E TTYHELLO Terminal Id error. id : *terminal_id*

Explanation: This message is written to the error log. The terminal ID was found to be outside the expected range. It is either negative or greater than the maximum number of Telnet server connections.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -3.

User or Operator Response: None.

System Programmer Response: Make sure that the terminal ID is properly specified. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TTYHELLO

EZY4209E TTYHELLO portno empty.

Explanation: This message is written to the error log. There is no free port number. The Telnet server exits.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -4.

User or Operator Response: None.

System Programmer Response: Make sure that the TCPIP configuration data set contains a valid PORT statement for the Telnet server. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TTYHELLO

EZY4210I KEY-IN YOUR TERMINAL TYPE & CONVERSION TYPE:

Explanation: This message prompts the user to enter the terminal type being used and the conversion type wanted.

System Action: Telnet continues.

User or Operator Response: Enter your terminal type and conversion type.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: ASKTTYTYPE,CHEKTTYTYPE

EZY4211E ASKTTYTYPE CodeConversion error. rc : rc

Explanation: This message is written to the error log. A code conversion error occurred during a call to the procedure ASKTTYTYPE. The return code in this message displays the offset in the current input buffer at which the error occurred.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -2.

User or Operator Response: None.

System Programmer Response: Check the input data to determine why this error occurred, and respond as indicated.

Source Data Set: TNDTRSVB

Procedure Name: ASKTTYTYPE

EZY4212E ASKTTYTYPE portno not found. portno : port_number

Explanation: This message is written to the error log. A free port number was not found.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -1.

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

System Programmer Response: Make sure that the TCPIP configuration data set contains a valid PORT statement for the Telnet server. For more information, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: ASKTTYTYPE

EZY4213E ASKTTYTYPE Terminal Id error. Id : terminal_ID

Explanation: This message is written to the error log. The terminal ID was outside the expected range. It is either negative or greater than the maximum number of Telnet server connections.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. Return code is set to -3.

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

System Programmer Response: Make sure that the terminal identifier is properly specified in the TCPIP configuration data set. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: ASKTTYTYPE

EZY4214E UNKNOWN TERMINAL TYPE.

Explanation: The user specified an unsupported terminal type. The supported options are TTY, VT100, and VT282

System Action: No Telnet session is established. TCPIP continues.

User or Operator Response: Enter a valid terminal type.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: CHEKTYPE

EZY4215E CHEKTYPE CodeConversion error. rc : rc

Explanation: A code conversion error occurred during a call to the procedure CHEKTYPE. The return code value displayed in this message indicates the offset in the current input buffer at which the error occurred.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -2.

User or Operator Response: None.

System Programmer Response: Check the input data to determine why this error occurred.

Source Data Set: TNDTRSVB

Procedure Name: CHEKTYPE

EZY4216E UNKNOWN CONVERSION TYPE.

Explanation: The user entered an unsupported conversion type. Supported values are: SFJ, J8N J8O, DEC, EUC, KSH, HAN, and TCH.

System Action: No Telnet session is established. TCPIP continues.

User or Operator Response: Enter a valid conversion type.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: CHEKTYPE

EZY4217E UNKNOWN TERMINAL & CONVERSION TYPE.

Explanation: The user entered an unsupported terminal type and an unsupported conversion type. Supported Conversion types are SFJ, J8N J8O, DEC, EUC, KSH, HAN, and TCH. Supported Terminal types are TTY, VT100, or VT282.

System Action: No Telnet session is established. Telnet continues.

User or Operator Response: Enter a valid terminal type and valid conversion type.

Source Data Set: TNDTRSVB

Procedure Name: CHEKTYPE

EZY4218E CHEKTYPE Terminal Id error. Id : *terminal_ID*

Explanation: The terminal ID was outside expected range. It is either negative or greater than the maximum number of Telnet server connections.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -3.

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

System Programmer Response: Make sure that the terminal identifier is properly specified in the TCPIP configuration data set. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

EZY4219E • EZY4223E

Source Data Set: TNDTRSVB

Procedure Name: CHEKTYPE

EZY4219E TERMINIT data allocation error.

Explanation: A nonzero return code was received from the subroutine DtAlloc, which is used to allocate storage for Telnet data. An attempt was made to allocate a data buffer the size of SIMDATASIZE for conversion codes. The return code is set to -5.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -5.

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

System Programmer Response: Make sure that enough storage is available to Telnet to allocate this data set.

Source Data Set: TNDTRSVB

Procedure Name: TERMINIT

EZY4220E TERMINIT address table error. data : *address_table*

Explanation: The address table is expected to be 0 prior to allocation. This is not the case. Storage might not have been freed.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -6.

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

System Programmer Response: Flush the address table and restart Telnet.

Source Data Set: TNDTRSVB

Procedure Name: TERMINIT

EZY4221E TERMINIT Terminal Id error. Id : *terminal_ID*

Explanation: The terminal ID was outside the expected range. It is either negative or greater than the maximum number of Telnet server connections.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -3.

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

System Programmer Response: Make sure that the terminal identifier is properly specified in the TCPIP configuration data set. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TERMINIT

EZY4223E MAPTOGRF Terminal Id error. Id : *terminal_ID*

Explanation: The terminal ID was outside the expected range. It is either negative or greater than the maximum number of Telnet server connections.

System Action: Control is returned to the TNSTMAS routine, where the session is terminated. The return code is set to -3.

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

System Programmer Response: Make sure that the terminal identifier is properly specified in the TCPIP configuration data set. For more information, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: MAPTOGRF

EZY4225I +++ MAPTOGRF +++

EZY4226E Abnormal data in TransfTermFlag

Explanation: This message is written to the error log. The variable TransfTermFlag is not an expected value. The following are expected values: READPAT, READMODA, READMOD, READBUFF.

System Action: TransfTermBufl is reset to zero. Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: MAPTOGRF

EZY4225I +++ MAPTOGRF +++

EZY4227I **TransfTermFlag** : *TransfTerm Flag*

EZY4228I **Output Data Length** : *TransfTerm Buffer length*

EZY4229I **Output Data Term Buffer**

Explanation: These messages are written to the error log. These messages display internal trace information.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: MAPTOGRF

EZY4230I +++ GRFTOMAP +++ **RCV DATA LENGTH** = *length*

Explanation: The function GRFTOMAP, which is used to map conversion characters, is starting. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4231I +++ GRFTOMAP +++ **RCV DATA TBUF**

Explanation: This message is displayed if the TRACE option is specified. This message displays trace information from the function GRFTOMAP, which is used to map conversion characters. Valid values are:

- EW, EWA--> Erase/Write order
- WRT--> Write order
- EAU--> Erase unprotected area
- RB--> Read Buffer
- RM--> Read modify
- RMA--> Read modify all
- WSF--> Write structured field

System Action: Telnet continues.

EZY4232I • EZY4235I

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4232I EWorder EXEC LENGTH = *Dleng*

Explanation: This message occurs during processing of the ERASE/WRITE internal 3270 order. If *Dleng* is negative, sense code 08002 is set, indicating an ERASE/WRITE or ERASE/WRITE alternate command. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: Fix the user program and retry it.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4233I WRTorder EXEC LENGTH = *Dleng*

Explanation: This message occurs during processing of the WRITE internal 3270 order. If *Dleng* is negative, sense code 08003 is set, indicating insufficient data in the WRITE command. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: Fix the user program and retry it.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4234I EAUorder EXEC

Explanation: This message occurs during processing of the ERASE internal 3270 order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4235I RBorder EXEC

Explanation: This message occurs during processing of the Read Buffer internal 3270 order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4236I RMorder EXEC

Explanation: This message occurs during processing of the Read Modify internal 3270 order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4237I RMAorder EXEC

Explanation: This message occurs during processing of the Read Modify All internal 3270 order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4238I WSForder EXEC LENGTH = *length*

Explanation: This message occurs during processing of the Write Structured Field internal 3270 order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

**EZY4240E ++ GRFTOMAP ABNORMAL END !! ++ : SCode : NAddr *SenseCode, ComBuffer Buffer Address*
*ComBuffer.DataBuffer ComBuffer.AttrBuffer***

Explanation: An error was encountered in the function GRFTOMAP, which is used to map conversion characters. The return code is set to -1.

System Action: Telnet 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 list of sense codes in the *z/OS Communications Server: IP User's Guide and Commands* to determine the cause of the error, and respond as indicated.

Source Data Set: TNDTRSVB

Procedure Name: GRFTOMAP

EZY4242E TELNET SERVER ERROR. SENSE CODE : *SenseCode*

Explanation: An error has been found during processing. The associated Sense code is printed. A list of the sense codes can be found in the *z/OS Communications Server: IP User's Guide and Commands*.

System Action: Telnet continues.

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

EZY4243I • EZY4246E

System Programmer Response: Use the sense code displayed in this message and the list of sense codes in the *z/OS Communications Server: IP User's Guide and Commands* to determine the cause of the error and respond as indicated.

Source Data Set: TNDTRSVB

Procedure Name: TTYGDBYE

EZY4243I IBM TCP/IP TELNET SERVER DBCS SERVICE END AT *time* ON *date*

Explanation: This message displays the time and date at which the Telnet session ended.

System Action: Telnet ends normally. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTRSVB

Procedure Name: TTYGDBYE

EZY4244E TTYGDBYE CodeConversion error. rc: *rc*

Explanation: A code conversion error occurred during a call to the procedure TTYGDBYE. The return code value displayed in this message indicates the offset in the current input buffer at which the error occurred.

System Action: Control is returned to the calling routine TNSTMAS and the session is terminated. The return code is set to -2.

User or Operator Response: None.

System Programmer Response: Check the input data to determine why the error occurred.

Source Data Set: TNDTRSVB

Procedure Name: TTYGDBYE

EZY4245E TTYGDBYE portno not found. portno : *port number*

Explanation: The indicated port number was not found.

System Action: Control is returned to the calling routine TNSTMAS and the session is terminated. The return code is set to -1.

User or Operator Response: Reissue the TELNET command specifying a valid port number. For more information about the TELNET command, refer to the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: If the error persists, make sure that the TCPIP configuration data set for the client contains a valid PORT statement for the Telnet server. For more information, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TTYGDBYE

EZY4246E TTYGDBYE Terminal Id error. id : *terminal ID*

Explanation: The terminal ID was outside the expected range. It is either negative or greater than the maximum number of Telnet server connections.

System Action: Control is returned to the calling routine TNSTMAS and the session is terminated. The return code is set to -3.

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

System Programmer Response: Make sure that the terminal identifier is properly specified in the client's TCPIP configuration data set. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TNDTRSVB

Procedure Name: TTYGDBYE

EZY4300I *** WRTorder DATA LENGTH = *length***

Explanation: This message displays the length of data being passed to the function WRTorder, which is used to write data to the terminal. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4301I ##### SFsbuodr EXEC

Explanation: This message displays trace information indicating that the user is currently in the Start Field branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4302I ##### SFEsbuodr EXEC

Explanation: This message displays trace information indicating that the user is currently in the Start Field Extent branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4303I ##### MFsbuodr EXEC

Explanation: The user is in the Modify Field branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4304I ##### SBAsbuodr EXEC

Explanation: The user is in the Set Buffer Address branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4305I ##### ICsbuodr EXEC

Explanation: The user is in the Insert Cursor branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4306I ##### PTsbuodr EXEC

Explanation: The user is in the Program Tab branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4307I ##### RAsbuodr EXEC

Explanation: The user is in the Repeat Address branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4308I ##### EUAsbuodr EXEC

Explanation: The user is in the Erase Unprotected to Address branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4309I SA EXEC

Explanation: This message occurs during processing of the Set Attribute internal 3270 order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4310I WriteData EXEC LENGTH = *length*

Explanation: The user is in the Normal Character branch of a case statement. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WRTorder

EZY4311I IDcnt = *ID count*

Explanation: This message displays the value of IDcount, a counter used in processing of the Write Structured Field order. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WSForder

EZY4312I READ PARTITION

Explanation: A Read Partition request is being processed. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WSForder

EZY4313I ErsRstID EXEC *i = i*

Explanation: An Erase/Reset request is being processed. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WSForder

EZY4314I SetRepID EXEC *i = i*

Explanation: The Telnet server is in set reply mode. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

EZY4315I • EZY4318I

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WSForder

EZY4315I OutbDtlID EXEC i = i

Explanation: This message indicates the amount of data being sent to a 3270-type terminal. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: WSForder

EZY4316I OutbDtlID (WRTorder) Exec Dataleng = Data length

Explanation: This message indicates the amount of data being passed to the function WRTorder, which is used to write data to a terminal. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: OutBdtID

EZY4317I OutbDtlID (EWorder) Exec Dataleng = Data length

Explanation: This message displays the amount of data being passed to the function EWorder. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: OutBdtID

EZY4318I OutbDtlID (EAUorder) EXEC

Explanation: This message indicates the function EAUorder is being called. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: OutBdtID

EZY4319I WRTorder error found with data *procedure_name*

Explanation: The DBCS transform function found incorrect data in the data stream. The buffer containing the incorrect data could not be transformed.

In the message text:

procedure_name

The name of the procedure where the incorrect data was found.

System Action: The transform of this buffer stops. The DBCS transform function continues with the next buffer.

User or Operator Response: Correct the error and retry the transform.

System Programmer Response: None.

Source Data Set: TND3270

Procedure Name: EWorder, OUTbDtID

EZY4350I +++ BuffShift (Buffer Shift) +++

EZY4351E First Parameter is Out of Range

Explanation: A request was made to shift data in a buffer, but the address passed is out of range. This message is written to the error log.

System Action: No data is moved. Control is returned to the calling routine and processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: BUFFSHIFT

EZY4350I +++ BuffShift (Buffer Shift) +++

EZY4352I Second Parameter is Out of Range

Explanation: A request was made to shift data in a buffer, but the address passed is out of range. This message is written to the error log.

System Action: No data is moved. Control is returned to the calling routine and processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: BUFFSHIFT

EZY4350I +++ BuffShift (Buffer Shift) +++

EZY4353I Third Parameter is Out of Range

Explanation: A request was made to shift data in a buffer, but the address passed is out of range.

System Action: No data is moved. Control is returned to the calling routine and processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: BUFFSHIFT

EZY4354W +++ OpenCodeTable : warning +++

EZY4356I ConType= *Con Type* ---> IBM *

Explanation: Either no conversion type was specified or an incorrect conversion type was found. Valid options are: J7O, J7N, J8O, J8N, A7O, A7N, A8O, A8N, SFJ, DEC, EUC, IBM, KSH, HAN, and TCH. This message is written to the error log.

System Action: The conversion type is set to IBM, and Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: OpenCodeTable

EZY4357W +++ CodeConversion : warning +++

EZY4358I ConDir= *Con Dir* ---> 0 *

Explanation: An unexpected value was found for ConDir.

System Action: The value is reset to 0, and Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: CodeConversion

EZY4357W +++ CodeConversion : warning +++

EZY4359I InState= *InState* ---> 0 *

Explanation: An unexpected value was found for InState.

System Action: The value is reset to 0, and Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: CodeConversion

EZY4357W +++ CodeConversion : warning +++

EZY4360I OutState= *OutState* ---> 0 *

Explanation: An unexpected value was found for OutState.

System Action: The value is reset to 0, and Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: CodeConversion

EZY4357W **+++ CodeConversion : warning +++**

EZY4361I **ConMode= *ConMode* ---> 0 ***

Explanation: An unexpected value was found for ConMode.

System Action: The value is reset to 0, and Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: CodeConversion

EZY4362W **+++ CloseCodeTable : warning +++**

EZY4356I **ConType= *Con Type* ---> IBM ***

Explanation: Either no conversion type was specified or an incorrect conversion type was found. Valid options are: J7O, J7N, J8O, J8N, A7O, A7N, A8O, A8N, SFJ, DEC, EUC, IBM, KSH, HAN, and TCH.

System Action: The conversion type is set to IBM, and Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: CloseCodeTable

EZY4363I ***CCTCLOS***

EZY4364I **ists= *ists***

Explanation: This message displays trace information. This message is displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMISCL

Procedure Name: CloseCodeTable.

EZY4366E **TNDBCSTM - Telnet DBCS Transform Mode - CodeFile DDName not found - *DD name***

Explanation: The Telnet server was unable to allocate ddname TNDBCSSL. This is the Telnet 3270 DBCS Transform data set containing binary translation table code files. This message is written to the error log.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: Update the JCL or TCPIP procedure to include TNDBCSTM.

Source Data Set: TNDMISCL

Procedure Name: RdCodeFiles

EZY4380I +++ ASCII TO MAP (ATMDEC) +++

EZY4381I Input Data Length : *Terminal Buffer length*

EZY4382I Input Data *Term Buffer*

Explanation: This message displays trace information about the TERMBUF buffer, which contains data mapping the terminal keys to hexadecimal characters. All of these key comparisons are declared as constants in the TNDCOMM macro. These messages are displayed if the TRACE option is specified.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDATMAP

Procedure Name: ATMDEC

EZY4400I ***** MAP to ASCII *****

EZY4401I TransfTermBuff : Set Data Length Over

EZY4402E Error Position No. is *sequence number*

EZY4403I Data Length is *length*

Explanation: An error was found after calling the internal routine CheckBufl. The sequence number specified corresponds to the position in the routine where the failure occurred. These messages are written to the error log.

System Action: Telnet ends normally. TCPIP continues.

User or Operator Response: None.

System Programmer Response: Save all trace information for problem determination and contact the IBM Software Support Center.

Source Data Set: TNDMTASC

Procedure Name: SetERRlog

EZY4400I ***** MAP to ASCII *****

EZY4404I Kanji Code Conversion : *rt = rc*

Explanation: An error occurred during code conversion while mapping input data to ASCII characters. The return code value displayed in this message indicates the offset in the current input buffer at which the error occurred. These messages are written to the error log.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: Check the input data to determine why this error occurred. and respond as indicated.

Source Data Set: TNDMTASC

Procedure Name: CharConv

EZY4400I ***** MAP to ASCII *****

EZY4405I Alpha Code Conversion : *rt = rc*

Explanation: An error occurred during code conversion while mapping input data to ASCII characters. The return code value displayed in this message indicates the offset in the current input buffer at which the error occurred. These messages are written to the error log.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: Check the input data to determine why this error occurred.

Source Data Set: TNDMTASC

Procedure Name: CharConv

EZY4400I ***** MAP to ASCII *****

EZY4406E Compare Error Address : *Cradr*

EZY4407I WritBuffer Data : *Dump of DataBuffer*

EZY4408I AttrBuffer Data : *Dump of AttrBuffer*

EZY4409I Disp.WritBuffer Data : *Dump of DispBuffer.DataBuffer*

EZY4410I Disp.AttrBuffer Data : *Dump of DispBuffer.Attrbuffer*

Explanation: These messages display trace information from the CodeConversion function. These messages are written to the error log.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMTASC

Procedure Name: DTCOMP

EZY4400I ***** MAP to ASCII *****

EZY4411I Alarm Code Conversion : *rt = rc*

Explanation: An error occurred during code conversion while mapping input data to ASCII characters. The return code value displayed in this message indicates the offset in the current input buffer at which the error occurred.

System Action: Telnet continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Check the input data to determine why this error occurred.

Source Data Set: TNDMTASC

Procedure Name: MTADEC

EZY4400I *** MAP to ASCII *******

EZY4412I Output Data Length : *length*

EZY4413I Output Data *dump of TermBuff*

Explanation: These messages display information about the contents of TermBuff, which is used to hold information mapping terminal keys to hexadecimal characters. These messages are written to the error log.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMTASC

Procedure Name: MTADEC

EZY4414I DataBuffer *dump of DataBuffer*

EZY4415I AttrBuffer *dump of AttrBuffer*

Explanation: Displayed is the value of the data and attribute buffers. These messages display during trace and are written to the error log.

System Action: Telnet continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDMTASC

Procedure Name: MTADEC

EZY4450I +++ CursorUp +++

EZY4451E Cursor Address is Out of Range. Non Process End.

Explanation: The placement of the cursor exceeded the supported buffer size of 1920. These messages are written to the error log.

System Action: The process ends normally.

User or Operator Response: None.

System Programmer Response: Make sure that the terminal type is properly specified and that the terminal is supported by Telnet.

Source Data Set: TNDTERM

Procedure Name: CursorUp

EZY4453I +++ CursorDown +++

EZY4451E Cursor Address is Out of Range. Non Process End.

Explanation: The placement of the cursor exceeded the supported buffer size of 1920. These messages are written to the error log.

System Action: The process ends normally.

User or Operator Response: None.

System Programmer Response: Make sure that the terminal type is properly specified and that the terminal is supported by Telnet.

Source Data Set: TNDTERM

Procedure Name: CursorDown

EZY4454I +++ CursorLeft +++

EZY4451E Cursor Address is Out of Range. Non Process End.

Explanation: The cursor placement exceeded the supported size of 1920 bytes. These messages are written to the error log.

System Action: The process ends normally.

User or Operator Response: None.

System Programmer Response: Make sure that the terminal type was properly specified and that the terminal is supported by Telnet.

Source Data Set: TNDTERM

Procedure Name: CursorLeft

EZY4455I +++ CursorRight +++

EZY4451E Cursor Address is Out of Range. Non Process End.

Explanation: The placement of the cursor exceeded the supported length of 1920.

System Action: The process ends normally.

User or Operator Response: None.

System Programmer Response: Make sure that the terminal type is properly specified and that the terminal is supported by Telnet.

Source Data Set: TNDTERM

Procedure Name: CursorRight

EZY4456E +++ CursHome +++ Home Position Address is Out of Range. Home Position is set to 0 for Next process.

Explanation: The cursor is not found at the home position as expected. This message is written to the error log.

System Action: The cursor is reset to 0 and processing continues.

User or Operator Response: Reset the cursor position if necessary.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: CursHome

EZY4459E +++ Reset Key +++

EZY4451E Cursor Address is Out of Range. Abnormal end.

Explanation: The cursor was placed out of the supported range of 1920. This message is written to the error log.

System Action: Processing continues, the return code is set to 1.

User or Operator Response: Reposition the cursor within the accepted range of 0 to 1920.

System Programmer Response: None.

EZY4461I • EZY4451E

Source Data Set: TNDTERM

Procedure Name: ResetKey

EZY4461I +++ FieldMark +++

EZY4451E **Cursor Address is Out of Range.**

Explanation: The cursor was placed beyond the supported range of 1920.

System Action: Processing continues.

User or Operator Response: Reposition the cursor within the accepted range of 0 to 1920.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: FieldMark

EZY4461I +++ FieldMark +++

EZY4463E **Attribute Address is Out of Range.**

Explanation: The given attribute address is out of the valid range.

System Action: Control is returned to the calling routine and processing continues.

User or Operator Response: Reenter the command specifying a valid attribute.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: FieldMark

EZY4461I +++ FieldMark +++

EZY4464I **Formatted Buffer.**

EZY4472E **Attribute Undefined.**

Explanation: The function SearchAttr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: Control is returned to the calling routine.

User or Operator Response: Resubmit the last request specifying valid attributes. For more information, refer to the *z/OS Communications Server: IP User's Guide and Commands*.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: FieldMark

EZY4465I +++ FieldTab +++

EZY4451E **Cursor Address is Out of Range. Address is set to 0 for normal end.**

Explanation: The cursor was placed outside the valid range of 0 to 1920.

System Action: The cursor address is set to 0 and processing continues.

User or Operator Response: Reposition the cursor if necessary.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: FieldTab

EZY4468I +++ FieldBackTab +++

EZY4451E **Cursor Address is Out of Range. Address is set to 0 for normal end.**

Explanation: The cursor was placed outside the supported range of 0 to 1920.

System Action: The cursor address is set to 0 and processing continues.

User or Operator Response: Reposition the cursor if necessary.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: FieldBackTab

EZY4469I +++ EraseEOF +++

EZY4451E **Cursor Address is Out of Range.**

Explanation: The cursor was placed outside the supported range of 0 to 1920.

System Action: Processing continues.

User or Operator Response: Reposition the cursor if necessary.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: EraseEOF

EZY4469I +++ EraseEOF +++

EZY4464I **Formatted Buffer.**

EZY4463E **Attribute Address is Out of Range.**

Explanation: The function SearchAttr, which is used to check conversion attributes, encountered an incorrect attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request, specifying a valid attribute.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: EraseEOF

EZY4469I +++ EraseEOF +++

EZY4464I **Formatted Buffer.**

EZY4472E **Attribute Undefined.**

Explanation: The function SearchAttr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: The request is not processed. Telnet continues.

EZY4473I • EZY4478I

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: EraseEOF

EZY4473I +++ EraseInput +++

EZY4464I Formatted Buffer.

EZY4472E Attribute Undefined.

Explanation: The function SearchAttr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: EraseInput

EZY4474E +++ AttPAKey +++ PA key ASCII Code Not found

Explanation: The PA key used is unsupported. Supported values are:

- PA1 (hex 6C for Ebcidic hex 31 for ASCII)
- PA2 (hex 6E for Ebcidic hex 32 for ASCII)
- PA3 (hex 6B for Ebcidic hex 33 for ASCII)

All other keys are ignored. The return code is set to 1.

System Action: The system returns to caller and continues processing.

User or Operator Response: Specify the correct PA key.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: AttPAKey

EZY4476I +++ AttPFKey +++

EZY4477E PF key No. out of Range

Explanation: The PF key used is unsupported. Supported values are PF1—PF24. The return code is set to 1.

System Action: Control returns to the caller and processing continues.

User or Operator Response: Specify a valid PF key.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: AttPFKey

EZY4478I +++ NextCurs +++

EZY4451E Cursor Address is Out of Range.

Explanation: The cursor was placed outside the supported range of 0 to 1920.

System Action: Processing continues.

User or Operator Response: Reposition the cursor within the supported range.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: NextCurs

EZY4478I +++ NextCurs +++

EZY4464I Formatted Buffer.

EZY4472E Attribute Undefined.

Explanation: The internal subroutine SearchAddr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: NextCurs

EZY4479I +++ Master Reset +++

EZY4451E Cursor Address is Out of Range. Non Process End.

Explanation: The cursor was placed outside the supported range of 0 to 1920.

System Action: Control is returned to the calling routine.

User or Operator Response: Reposition the cursor within the supported range.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: MstReset

EZY4480I +++ Dupric +++

EZY4451E Cursor Address is Out of Range.

Explanation: The cursor was placed outside of the supported range of 0 to 1920.

System Action: Control is returned to the calling routine.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: DUPRIC

EZY4480I • EZY4451E

EZY4480I **+++ Dupric +++**

EZY4464I **Formatted Buffer.**

EZY4463E **Attribute Address is Out of Range.**

Explanation: The internal subroutine SearchAttr, which is used to check conversion attributes, encountered an attribute that was out of range.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: DUPRIC

EZY4480I **+++ Dupric +++**

EZY4464I **Formatted Buffer.**

EZY4472E **Attribute Undefined.**

Explanation: The internal subroutine SearchAttr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: DUPRIC

EZY4481I **+++ CrgReturn +++**

EZY4451E **Cursor Address is Out of Range.**

Explanation: The cursor was placed outside the supported range of 0 to 1920. The return code is set to 1.

System Action: Control is returned to the calling routine.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: CrgReturn

EZY4482I **+++ NormalChar (Input) +++**

EZY4451E **Cursor Address is Out of Range.**

Explanation: The cursor was placed outside the supported range of 0 to 1920.

System Action: Control returns to the calling routine.

User or Operator Response: Reposition the cursor within the supported range.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: NormalChar

EZY4482I +++ NormalChar (Input) +++

EZY4464I Formatted Buffer.

EZY4463E Attribute Address is Out of Range.

Explanation: The internal subroutine SearchAttr, which is used to check conversion attributes, encountered an out of range attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: NormalChar

EZY4482I +++ NormalChar (Input) +++

EZY4464I Formatted Buffer.

EZY4472E Attribute Undefined.

Explanation: The internal subroutine SearchAttr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: NormalChar

EZY4483I +++ Delete1C +++

EZY4451E Cursor Address is Out of Range.

Explanation: The cursor was placed outside the supported range of 0 to 1920.

System Action: Control is returned to the calling routine.

User or Operator Response: Reposition the cursor within the valid range.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: Delete1C

EZY4483I +++ Delete1C +++

EZ4464I Formatted Buffer.

EZY4472E Attribute Undefined.

Explanation: The internal subroutine SearchAttr, which is used to check conversion attributes, encountered an undefined attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: Delete1C

EZY4483I +++ Delete1C +++

EZY4464I Formatted Buffer.

EZY4463E Attribute Address is Out of Range.

Explanation: The internal subroutine SearchAttr, which is used to check conversion attributes, encountered an out of range attribute.

System Action: The request is not processed. Telnet continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: None.

Source Data Set: TNDTERM

Procedure Name: Delete1C

EZY4484I ++ Inputerr ++ message

Explanation: The Telnet DBCS code has detected an error with the input data. Some of the possible reasons are:

- The keyboard is locked
- Input was tried in a protected field
- The command was out of sequence
- The data entered was not valid.

An informational message is displayed in the variable text field of the message. The layout of the variable text field is described below:

xxxxxxxx-yy yy nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn

where

xxxxxx The translation routine being used. This routine can be correlated with the code page being used and the translation being done. The first few characters are the translation and the rest, if included, the codepage being used.

```

Translation
ATM - ASCII to 3270
TNDTERM - 3270 terminal translation

Codepage
SJF - SJISKANJI
J80 - JIS78KJ
J8N - JIS83KJ
EUC - EUCKANJI

```

DEC - DECKANJI
HAN - HANGEUL
KSC - KSC5601
TCH - TCHINESE

yyyyyy The command(s) being processed. There may be multiple commands strung together all separated by a hyphen (-) An example of this would be

NULL-ESC-CRTLK

nnnnnn The reason.

Possible reasons and actions are:

Reason Action

ATMDEC-CtrlA INPUT INHIBITED

The keyboard is locked. Reset the keyboard.

ATMDEC-NULL INVALID COMMAND RECEIVED

The previous command was not valid. Retry.

ATMDEC *x*

The command/data was either not valid or out of sequence.

System Action: TCPIP continues.

User or Operator Response: Resubmit the request specifying valid attributes.

System Programmer Response: From the trace message try to determine the cause of the error, if unable, then contact the IBM Software Support Center.

Source Data Set: TNDTERM TNDATMAP

Procedure Name: various

Chapter 6. EZY5xxxx messages

EZY5398E **REQUEST TO START TCP/IP VIA MVPMAIN IS REJECTED.**

Explanation: Program MVPMAIN was invoked with MODULE=TCPIP specified on the PARM= parameter of the JCL EXEC statement. The TCP/IP address space can no longer be started using program MVPMAIN.

System Action: The task initialization function is terminated.

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

System Programmer Response: See sample procedure TCPIPROC in the SEZAINST data set for the correct program name to be used for starting the TCP/IP address space.

Source Data Set: MVPMAIN

Procedure Name: Mainline code

Chapter 7. EZY6xxxx messages

EZY6001I EZAZSSI Already Active, Start Rejected

Explanation: EZAZSSI is already active, attempting to start TNF and/or VMCF.

System Action: This duplicate copy of EZAZSSI ends.

User or Operator Response: Either wait for VMCF and/or TNF to start, cancel then restart EZAZSSI, or ignore the message.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6002I TNF Already Active

Explanation: TNF is already active, therefore it is not started.

System Action: Start VMCF if needed, then EZAZSSI ends.

User or Operator Response: Either end TNF and restart, or ignore the message.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6003I VMCF Already Active

Explanation: VMCF is already active, therefore it is not started.

System Action: EZAZSSI ends.

User or Operator Response: Either end VMCF and restart, or ignore the message.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6004I No Action Taken

Explanation: VMCF and TNF are both running; there is nothing to start.

System Action: TNF/VMCF continue, EZAZSSI ends.

User or Operator Response: Either stop VMCF and/or TNF then restart EZAZSSI, or ignore the message.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6005I Invalid Start Sequence in Progress

Explanation: VMCF is up and running and TNF is down.

System Action: VMCF continues, EZAZSSI ends.

User or Operator Response: Stop VMCF and then restart EZAZSSI.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6006E Subsystem name in IEFSSNxx is not VMCF

Explanation: There is no VMCF entry in IEFSSNxx.

System Action: EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: Add VMCF entries to IEFSSNxx and re-IPL.

Source Data Set: EZAZSSI

EZY6007E VMCF Nodename must be specified

Explanation: VMCF requires that a nodename be specified at least once per IPL.

System Action: EZAZSSI ends.

User or Operator Response: Specify a nodename when starting EZAZSSI. Refer to the *z/OS Communications Server: IP Diagnosis Guide* for more information.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6008I VMCF Start Initiated

Explanation: A VMCF address space start was issued.

System Action: EZAZSSI continues; start of VMCF expected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6009E Refused to create VMCF address space

Explanation: The VMCF address space start from IEEMB881 failed.

System Action: EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: Check to see that the data sets containing the respective load modules are cataloged and available.

Source Data Set: EZAZSSI

EZY6010E VMCF Not Initialized, Processing Continues

Explanation: The VMCF address space start-up timed out. See the messages that usually accompany this message for an indication of the actual problem.

System Action: EZAZSSI ends.

User or Operator Response: Determine why VMCF did not start and restart EZAZSSI.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6011I VMCF Initialization Complete

Explanation: VMCF is active and available. VMCF applications can be started.

System Action: VMCF continues, EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6012E Error creating PC numbers, is LPA correct?

Explanation: PC numbers could not be created.

System Action: TNF/VMCF ends, EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: This is probably an installation error. Check the LPA data sets.

Source Data Set: EZAZSSI

EZY6013E VMCF ended: Error in initialization

Explanation: An error was encountered in initialization. See the messages that usually accompany this message for an indication of the actual problem.

System Action: VMCF ends.

User or Operator Response: Dependent on accompanying messages.

System Programmer Response: Dependent on accompanying messages.

Source Data Set: EZAZSSI

EZY6014E Subsystem name in IEFSSNxx is not TNF

Explanation: There is no TNF entry in IEFSSNxx.

System Action: EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: Add TNF entry to IEFSSNxx and re-IPL.

Source Data Set: EZAZSSI

EZY6015I TNF Start Initiated

Explanation: A TNF address space start was issued.

System Action: EZAZSSI continues; start of TNF expected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6016E Refused to create TNF address space

Explanation: The TNF address space start from IEEMB881 failed.

System Action: EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: Check to see that the data sets containing the respective load modules are cataloged and available.

Source Data Set: EZAZSSI

EZY6017E TNF Not Initialized, Processing Continues

Explanation: The TNF address space start-up timed out. See the messages that accompany this message for an indication of the actual problem.

System Action: EZAZSSI ends.

User or Operator Response: Determine why TNF did not start and restart EZAZSSI.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6018I TNF Initialization Complete

Explanation: TNF is active and available. TNF applications can be started.

System Action: EZAZSSI will attempt to start VMCF.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZSSI

EZY6019E TNF ended: MVPTRML Resource Manager Not Established

Explanation: A problem exists in the RESMGR interface.

System Action: TNF ends, EZAZSSI ends.

User or Operator Response: None.

System Programmer Response: This is probably an installation error. This message can be issued if resource manager load module MVPTRML is not found in LPA. Ensure that the TCP/IP LPA load library SEZALPA is defined in one of the LPALSTxx members of SYS1.PARMLIB.

Source Data Set: EZAZSSI

EZY6020I TNF not started correctly

Explanation: TNF was started by a method other than EZAZSSI or EZAZSSI timed out because it took too long to start TNF.

System Action: TNF ends.

User or Operator Response: Use EZAZSSI to start TNF.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6021I TNF Stop Rejected - Active Users Exist

Explanation: Active users still exist for TNF; the stop is ignored.

System Action: TNF continues.

User or Operator Response: Either end the user address spaces or use the **F TNF,REMOVE...** command to remove the users from the table.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6022I TNF Stop Accepted

Explanation: The TNF STOP command was accepted.

System Action: TNF stops.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6023I TNF Modify Display Accepted

Explanation: The Modify Display command for TNF was accepted.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6024I TNF Modify Remove Accepted

Explanation: The Modify Remove command for TNF was accepted.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: This is probably an installation error. This message can be issued if resource manager load module MVPTRML is not found in LPA. Ensure that the TCP/IP LPA load library SEZALPA is defined in one of the LPA1STxx members of SYS1.PARMLIB.

Source Data Set: MVPTNF

EZY6025I TNF Modify Command Not Recognized

Explanation: The Modify command for TNF was not understood.

System Action: TNF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6026I TNF Start Accepted

Explanation: The TNF Start command was accepted.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6027I TNF Command Not Recognized

Explanation: The command entered for TNF was not understood.

System Action: TNF continues.

User or Operator Response: For information on TNF commands, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

EZY6028I • EZY6032I

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6028I TNF REMOVE not valid

Explanation: The Modify remove command for TNF was not understood.

System Action: TNF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6029I TNF REMOVEXXXXXX not valid

Explanation: The Modify remove command for TNF was not understood.

System Action: TNF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6030I TNF All Users Removed

Explanation: The Modify remove,name=* was executed successfully.

System Action: TNF continues.

User or Operator Response: TNF can be ended now.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6031I TNF User *userid* Not Found

Explanation: The Modify remove/display,name=*userid* was not found in the table.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6032I TNF User *userid* Removed

Explanation: The Modify remove,name=*userid* was found and removed.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6033I TNF DISPLAY not valid

Explanation: The Modify display command for TNF was not understood.

System Action: TNF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6034I TNF DISPLAYXXXXXX not valid

Explanation: The Modify display command for TNF was not understood.

System Action: TNF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6035I TNF User *userid* Not Found

Explanation: The Modify remove,name=*userid* for TNF was not found in the table.

System Action: TNF continues.

User or Operator Response: Do a modify display to see if the user still exists.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6036I TNF Active ASIDs

Explanation: This message is issued in response to the Modify display command. It is displayed when users exist.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6037I Name ASID

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6038I *userid asid*

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

EZY6039I • EZY6044E

Source Data Set: MVPTNF

EZY6039I End

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6040I No TNF users exist

Explanation: This message is issued in response to the Modify command. It is displayed when no users exist.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6041I TNF Modify Accepted

Explanation: The Modify command for TNF was accepted.

System Action: TNF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

EZY6042E TNF ended: MVPTTRML Resource Manager Not Established

Explanation: A problem exists in the RESMGR interface.

System Action: TNF ends.

User or Operator Response: None.

System Programmer Response: This is probably an installation error.

Source Data Set: MVPTSSI

EZY6043I TNF Start Initiated

Explanation: A TNF address space start was issued.

System Action: EZAZSSI continues; start of TNF expected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTSSI

EZY6044E Refused to create TNF address space

Explanation: The address space start from IEEMB881 failed.

System Action: IPL Continues.

User or Operator Response: None.

System Programmer Response: Check to see that the data sets containing the respective load modules are cataloged and available.

Source Data Set: MVPTSSI

EZY6045E TNF Not Initialized, Processing Continues

Explanation: The TNF address space start-up timed out. See the messages that usually accompany this message for an indication of the actual problem.

System Action: IPL continues.

User or Operator Response: None.

System Programmer Response: Correct the reason TNF did not start, and use EZAZSSI to restart TNF/VMCF.

Source Data Set: MVPTSSI

EZY6046I TNF Initialization Complete

Explanation: TNF is active and available.

System Action: TNF continues, IPL continues.

User or Operator Response: TNF applications can be started.

System Programmer Response: None.

Source Data Set: MVPTSSI

EZY6047E Subsystem name in IEFSSNxx is not TNF

Explanation: There is not a TNF entry in IEFSSNxx.

System Action: TNF ends.

User or Operator Response: None.

System Programmer Response: Add TNF entry to IEFSSNxx and re-IPL.

Source Data Set: MVPTSSI

EZY6048E VMCF ended: Node ID not specified in IEFSSNxx VMCF

Explanation: A nodename must be specified in the IEFSSNxx member for VMCF when using the migration implementation of restartable VMCF.

System Action: IPL continues.

User or Operator Response: Start EZAZSSI with a nodename after IPL completes.

System Programmer Response: For future IPLs add a nodename to the IEFSSNxx member.

Source Data Set: MVPXSSI

EZY6049I VMCF Start Initiated

Explanation: A VMCF address space start was issued.

System Action: Start of VMCF expected.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXSSI

EZY6050E Refused to create VMCF address space

Explanation: The address space start from IEEMB881 failed.

System Action: IPL continues.

User or Operator Response: Start EZAZSSI when the reason for the failure is corrected.

System Programmer Response: Check to see that the data sets containing the respective load modules are cataloged and available.

Source Data Set: MVPXSSI

EZY6051E VMCF Not Initialized, Processing Continues

Explanation: The VMCF address space start-up timed out. See the messages that usually accompany this message for an indication of the actual problem.

System Action: IPL continues.

User or Operator Response: Determine why VMCF did not start and start EZAZSSI when the IPL completes.

System Programmer Response: None.

Source Data Set: MVPXSSI

EZY6052I VMCF Initialization Complete

Explanation: VMCF is active and available.

System Action: VMCF continues.

User or Operator Response: VMCF applications can be started.

System Programmer Response: None.

Source Data Set: MVPXSSI

EZY6053E Error creating PC numbers, is LPA correct?

Explanation: PC numbers could not be created.

System Action: VMCF ends.

User or Operator Response: None.

System Programmer Response: This is probably an installation error. Check the LPA data sets.

Source Data Set: MVPXSSI

EZY6054E VMCF ended: Error in initialization

Explanation: An error was encountered in initialization. See the messages that usually accompany this message for an indication of the actual problem.

System Action: VMCF ends.

User or Operator Response: Dependent on accompanying messages.

System Programmer Response: Dependent on accompanying messages.

Source Data Set: MVPXSSI

EZY6055E Subsystem name in IEFSSNxx is not VMCF

Explanation: There is not a VMCF entry in IEFSSNxx.

System Action: IPL Continues.

User or Operator Response: None.

System Programmer Response: Add VMCF entries to IEFSSNxx and re-IPL.

Source Data Set: MVPXSSI

EZY6056I VMCF not started correctly

Explanation: TNF was started by a method other than EZAZSSI or EZAZSSI timed out because it took too long to start VMCF.

System Action: VMCF ends.

User or Operator Response: Use EZAZSSI to start TNF.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6057I VMCF Stop Accepted

Explanation: The VMCF command was accepted.

System Action: VMCF stops.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6058I VMCF Modify Accepted

Explanation: The Modify command for VMCF was accepted.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6059I VMCF Modify Display Accepted

Explanation: The Modify display command for VMCF was accepted.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6060I VMCF Modify Remove Accepted

Explanation: The Modify Remove command for VMCF was accepted.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6061I VMCF Start Accepted

Explanation: The VMCF Start command was accepted.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPTNF

Source Data Set: MVPXVI

EZY6062I VMCF Command Not Recognized

Explanation: The command entered for VMCF was not understood.

System Action: VMCF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6063I VMCF Modify Command Not Recognized

Explanation: The Modify command for VMCF was not understood.

System Action: VMCF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6064I VMCF Stop Rejected - Active Users Exist

Explanation: Active users still exist for VMCF; the stop is ignored.

System Action: VMCF continues.

User or Operator Response: Either terminate the user address spaces or use the **F TNF,REMOVE...** command to remove the users from the table.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6065I VMCF REMOVE not valid

Explanation: The Modify remove command was not understood.

System Action: VMCF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6066I VMCF REMOVEXXXXXX not valid

Explanation: The Modify remove command was not understood.

System Action: VMCF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6067I VMCF All Users Removed

Explanation: The Modify remove,name=* was executed successfully.

System Action: VMCF continues.

User or Operator Response: VMCF can be stopped now.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6068I VMCF User *userid* Not Found

Explanation: The Modify remove,name=*userid* was not found in the table.

System Action: VMCF continues.

User or Operator Response: Do a modify display to see if the user still exists.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6069I VMCF User *userid* Removed

Explanation: The Modify remove,name=*userid* was found and removed.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6070I VMCF DISPLAY Invalid

Explanation: The Modify display command was not understood.

System Action: VMCF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6071I VMCF DISPLAYXXXXXX not valid

Explanation: The Modify display command was not understood.

System Action: VMCF continues.

User or Operator Response: For information on the Modify command, refer to the *z/OS Communications Server: IP Diagnosis Guide*.

EZY6072I • EZY6077I

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6072I VMCF User *userid* Not Found

Explanation: The Modify remove/display,name=*userid* was not found in the table.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6073I VMCF Active ASIDs

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6074I Name ASID

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6075I *userid asid*

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6076I End

Explanation: This message is issued in response to the Modify command. It is displayed when users exist.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6077I No VMCF users exist

Explanation: This message is issued in response to the Modify command. It is displayed when no users exist.

System Action: VMCF continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MVPXVI

EZY6101I *fsswtr_id* **FSS=fss_addr, FSA=fsa_addr, FSD=fsd_addr, FSJ=fsj_addr**

Explanation: This message is issued in conjunction with message EZY6102I at FSS startup time. Listed for the FSS specified in *fsswtr_id* are the addresses of the FSS, FSA, FSD and FSJ modules.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6102I *fsswtr_id* **FSID=fsid, ASID=asid, SUB=subsystem_id, OUT=routeode**

Explanation: This message is issued in conjunction with message EZY6101I at FSS startup time. Listed for the FSS specified in *fsswtr_id* are the FSID, ASID, Subsystem ID, and Route Code values.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6110I *fsswtr_id* **OPTS: option_1,option_2,...**

Explanation: This message is issued at FSS startup time. Multiple occurrences of the message are issued to list all FSS startup options in effect.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6111I *fsswtr_id* **STATS: statistic_1,statistic_2...**

Explanation: This message is issued when the FSS is modified. Multiple occurrences of the message are issued to list some statistics that may be useful for problem analysis.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6112I *fsswtr_id* **FSS ENTERING WAIT**

Explanation: The FSS named in the message has entered a wait state.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6113I *fsswtr_id* **FSS POSTED BY FSA TASK**

Explanation: The FSS named in the message has been posted by an FSA.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6114I *fsswtr_id* **FSS POSTED BY JES**

Explanation: The FSS named in the message has been posted by JES.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6115I *fsswtr_id* **FSS POSTED BY JES, ORDID=*fss_order_id***

Explanation: The FSS named in the message has been posted by JES, with the ORDER identified in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6116I *fsswtr_id* **FSS POSTED BY COMMAND**

Explanation: The FSS named in the message has been posted by an MVS operator console command.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6117I *fsswtr_id* **START-FSS CONNECT FSIREQ SUCCESSFUL**

Explanation: The FSS-CONNECT request was successful for the FSS named in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6118I *fsswtr_id* **STOP-FSS DISCONNECT FSIREQ SUCCESSFUL**

Explanation: The FSS-DISCONNECT request was successful for the FSS named in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6119I *fsswtr_id* **START-FSA ATTACH SUCCESSFUL, FSID=*fss_id***

Explanation: An FSA was successfully attached to the FSS named in the message. The FSS ID is also displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6120I *fsswtr_id* **STOP-FSA DETACH SUCCESSFUL, FSID=*fss_id***

Explanation: An FSA was successfully detached from the FSS named in the message. The FSS ID is also displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6121I *fsswtr_id* **FSS-AMODE/JES-AMODE/RESOLVED-CB-RMODE=*fss_amodeljes_amodelcb_rmode***

Explanation: The control block RMODE that the FSS and FSA tasks will use (for the FSS named in the message) has been determined from the early FSS AMODE option setting. The RMODE setting will be permanent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6122I *fsswtr_id* **INIT FUNCTION DONE**

Explanation: The initialization functions have been successfully completed for the FSS named in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6131E *fsswtr_id* **START-FSS START TOKEN INVALID: *mgcr_token_addr***

Explanation: The MGCR token address is invalid. The FSS cannot be started.

System Action: Processing ends.

User or Operator Response: The FSS cannot be explicitly started by the operator or run as a batch job. The FSS is started by JES.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6132E *fsswtr_id* **START-FSS ERROR, NO START CIB FOUND**

Explanation: No START-verb CIB was found from the CSCB. This is an internal configuration error.

System Action: Processing ends.

User or Operator Response: Notify Systems Programming.

System Programmer Response: Contact IBM Support.

EZY6133E • EZY6137E

Source Data Set: EZAPPF5

EZY6133E *fsswtr_id* **START-FSS START CMD INVALID, SUBSYS=***subsystem_name*

Explanation: An error was encountered in the FSS START PARM. The subsystem name is invalid. This is an internal configuration error.

System Action: Processing ends.

User or Operator Response: Notify Systems Programming.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6134E *fsswtr_id* **START-FSS START CMD INVALID, FSID=***fss_id*

Explanation: An error was encountered in the FSS START PARM. The functional subsystem is invalid. This is an internal configuration error.

System Action: Processing ends.

User or Operator Response: Notify Systems Programming.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6135E *fsswtr_id* **START-FSS START CMD INVALID, ROUTCDE=***route_code*

Explanation: An error was encountered in the FSS START PARM. The FSS-level message route code is invalid. This is an internal configuration error.

System Action: Processing ends.

User or Operator Response: Notify Systems Programming.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6136E *fsswtr_id* **START-FSS PARMLIB DD OPEN FAILED**

Explanation: The data set defined by the PARMLIB DD JCL statement is known to exist but could not be opened.

System Action: Processing ends.

User or Operator Response: Determine why the data set cannot be opened.

System Programmer Response: None.

Source Data Set: EZAPPF5

EZY6137E *fsswtr_id* **LOAD OF EZAPPF5CM FAILED**

Explanation: The NPF file-create module could not be loaded.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Determine why LOAD failed for EZAPPF5CM.

Source Data Set: EZAPPF5

EZY6138E *fsswtr_id* INIT CALL FAILED IN FCM, RC=*return_code*

Explanation: The INIT function failed within the NPF file-create module.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6139E *fsswtr_id* START-FSS CONNECT FSIREQ FAILED, SSOBRETN=*fsi_return_code*

Explanation: The FSS_CONNECT FSIREQ call failed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6140E *fsswtr_id* START-FSS CONNECT FSIREQ FAILED, R15=*ssreq_return_code*

Explanation: The FSS_CONNECT FSIREQ call failed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6141E *fsswtr_id* STOP-FSS REFUSED, FSAS STILL UP

Explanation: The FSS cannot SHUTDOWN while active FSAs are still present. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6142E *fsswtr_id* STOP-FSS DISCONNECT FSIREQ FAILED, SSOBRETN=*fsi_error_return_code*

Explanation: The FSS did not DISCONNECT successfully. The return code from FSIREQ is listed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6143E *fsswtr_id* STOP-FSS DISCONNECT FSIREQ FAILED, R15=*register_15*

Explanation: The FSS did not DISCONNECT successfully. The return code from the Subsystem Interface is listed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

EZY6144E • EZY6148E

Source Data Set: EZAPPF5

EZY6144E *fsswtr_id* STOP-FSS ABNORMAL DISCONNECT REQUESTED

Explanation: The FSS did not DISCONNECT successfully. The return code from the Subsystem Interface is listed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6145E *fsswtr_id* START-FSA-SDTE GETMAIN FAILED

Explanation: GETMAIN was called for the FSA Daughter Task Element control block. The call failed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

Source Data Set: EZAPPF5

EZY6146E *fsswtr_id* START-FSA ECBLIST GETMAIN FAILED

Explanation: GETMAIN was called for the ECB list data area.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

Source Data Set: EZAPPF5

EZY6147E *fsswtr_id* START-FSA EP IDENTITY FAILED, R15=*identify_return_code*

Explanation: The entry point for the FSA could not be identified. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6148E *fsswtr_id* START-FSA SUBTASK ATTACH FAILED, R15=*attach_return_code*

Explanation: The FSA ATTACH failed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPF5

EZY6149E *fsswtr_id* **START-FSA ORDER REFUSED - sdte_fsid UP**

Explanation: The FSA-START ORDER was refused. The FSA is already started. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFS

EZY6150E *fsswtr_id* **STOP-FSA SUBTASK DETACH FAILED, R15=detach_return_code**

Explanation: The FSA DETACH failed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFS

EZY6151E *fsswtr_id* **STOP-FSA ORDER REFUSED - fsid NOT UP**

Explanation: The FSA STOP order was processed, but the FSID could not be found. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFS

EZY6152E *fsswtr_id* **UNSUPPORTED FSS ORDER, ORDID=fss_order_id**

Explanation: The FSS order could not be recognized or is invalid. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFS

EZY6160W *fsswtr_id* **STOP (P) COMMANDS ARE NOT SUPPORTED AT THIS TIME**

Explanation: The FSS cannot be stopped via the command input buffer (P *fsswtr_id*) in the current product release.

System Action: Processing continues.

User or Operator Response: If you wish to terminate the FSS, you must drain or stop the printer via the appropriate JES command.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6161W *fsswtr_id* **UNRECOGNIZED COMMAND RECEIVED. REFUSED**

Explanation: The command received from the console command input buffer is not recognized as a valid command.

System Action: Processing continues.

User or Operator Response: Re-enter the command.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6162W *fsswtr_id* **INVALID FSS OPTION NEAR:** *invalid_data*

Explanation: The option input is invalid.

System Action: Processing continues.

User or Operator Response: Re-input the option.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6163W *fsswtr_id* **JES ASSUMING AMODE=31, AMODE=24 NOW INVALID**

Explanation: The AMODE affecting existing tasks has been set to "31", and AMODE=24 cannot be used.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6164W *fsswtr_id* **FSA TASK TERMINATED, FSID=*fss_id***

Explanation: The FSS has received notification that the FSA with the FSID displayed in the message text has ended.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6165W *fsswtr_id* **FSA PRE-CONNECT FAILURE RESPONSE FSI-SENT, FSID=*fss_id***

Explanation: The FSS has sent notification via FSI SEND that the START FSA ORDER has failed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6166W *fsswtr_id* **UNABLE TO GET FSIP AREA ABOVE 16MEG**

Explanation: The FSS was unable to GETMAIN the needed storage area above the 16MB line.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFS

EZY6175I *fsswtr_id printer_id* **FSA FSID=*fsid*, UCB=*ucb_name*, AMODE=*addr_mode***

Explanation: This message is issued at FSA startup time. Listed for the FSA specified in *printer_id* for the FSS specified in *fsswtr_id* are the FSID, UCB, and AMODE.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6176I *fsswtr_id printer_id GDS job_number dd_name job_name separator_indicator*

Explanation: This message is issued when a new data set is received by the FSS's FSA from JES. Job number, DD name, and job name are displayed, along with a print data set separator page indicator defined as follows:

NOSEP

No separator page is required.

START A start data set separator page is required.

CONT A continue data set separator page is required.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFA

EZY6177I *fsswtr_id printer_id RDS job_number dd_name smf_indicator*

Explanation: This message is issued if there is a data set at the stacker when RELDS is about to be issued. An indicator is issued at the end of the message as follows:

INCOMPLETE

RELDS is incomplete.

SMF=YES

An SMF type 6 record was requested.

SMF=NO

No SMF type 6 record was requested.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFA

EZY6178I *fsswtr_id printer_id EOF/EOG job_number dd_name record_count TRLR/NOTRLR*

Explanation: This message is issued to indicate that the data set has been marked for end-of-file or end-of-group. JES job number, DD name, and record count are shown in the message. The message will show either 'EOF' or 'EOG' between *printer_id* and *job_number*. 'TRLR' will appear at the end of the message to acknowledge the last data set in a JES2 output group or a JES3 job.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFA

EZY6179I *fsswtr_id printer_id SND job_number dd_name OOP-TRACKING*

Explanation: This message indicates that a tracking FSI SEND operation has been performed. Job number and DD name appear in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFA

EZY6180I *fsswtr_id printer_id* **FSA ENTERING WAIT**

Explanation: The FSA associated with *printer_id* is waiting for work.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6181I *fsswtr_id printer_id* **FSA POSTED BY JES**

Explanation: The FSI Order/Post ECB of the FSA associated with (*printer_id*) has been posted by JES.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6182I *fsswtr_id printer_id* **FSA POSTED BY JES, ORDID=*fsa_order_id***

Explanation: The FSI Order/Post ECB of the FSA associated with (*printer_id*) has been posted by JES with the ORDER ID displayed in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6183I *fsswtr_id printer_id* **FSA POSTED BY NPRO TIMER POP**

Explanation: The NPRO STIMER ECB of the FSA associated with (*printer_id*) has been posted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6184I *fsswtr_id printer_id* **FSA POSTED BY FSS**

Explanation: The FSA associated with (*printer_id*) has been posted by the FSS.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6185I *fsswtr_id printer_id* **FSA DIRECTED BY FSS TO TERMINATE**

Explanation: The FSA associated with (*printer_id*) has been posted by the FSS and directed to terminate.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6186I *fsswtr_id printer_id* **START-FSA CONNECT FSIREQ SUCCESSFUL**

Explanation: The FSA-CONNECT FSIREQ request succeeded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6187I *fsswtr_id printer_id* **STOP-FSA DISCONNECT FSIREQ SUCCESSFUL**

Explanation: The FSA-DISCONNECT FSIREQ request succeeded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6188I *fsswtr_id printer_id* **DEVICE IS STARTING**

Explanation: The FSA device is starting and is available for work.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6189I *fsswtr_id printer_id* **DEVICE IS STOPPING**

Explanation: The FSA device is stopping and is not available for work.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6190I *fsswtr_id printer_id* **OPERATOR INTERVENTION ORDERED**

Explanation: The FSA device requires operator intervention. The FSA has received an asynchronous ORDER request, via an FSIREQ, requesting that the FSA flush its device buffer and prepare for something that probably involves operator setup.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6191I *fsswtr_id printer_id* **DEVICE SYNC ORDERED**

Explanation: The FSA has an asynchronous ORDER request via an FSIREQ requesting that the FSA synchronize its processing to the point of actual printing. That is, a JES command such as a device backspace or forward space has been issued and it should affect the dataset at the OOP (Operator Orientation Point).

System Action: Processing continues.

User or Operator Response: None.

EZY6192I • EZY6196I

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6192I *fsswtr_id printer_id* **DEVICE SET ORDERED**

Explanation: The FSA has an asynchronous ORDER request via an FSIREQ requesting that the FSA reset some parameters. An example of this would be a JES2 \$T command to change NPRO.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6193I *fsswtr_id printer_id* **OUT OF BUFFERS IN GETREC, TRYING AGAIN**

Explanation: The GETREC FSIREQ could not get the required number of buffers.

System Action: The request will be reissued.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6194I *fsswtr_id printer_id* **NOTIFY SENT TO USER=user_list**

Explanation: The user IDs listed in *user_list* were found in the NOTIFY parameter on the OUTPUT JCL statement. Data set processing results will be sent to the users in the list.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6195I *fsswtr_id printer_id* **DATA SET AT OOP RESCHEDULED**

Explanation: The data set at the Operator Orientation Point has been rescheduled.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6196I *fsswtr_id printer_id* **DATA SET AT OOP CANCELED**

Explanation: The data set at the Operator Orientation Point has been cancelled.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6197I *fsswtr_id printer_id* **NO DATA SET AT OOP FOR SYNCH/OPINT**

Explanation: A SYNCH or OPINT command was processed but no data set exists at the Operator Orientation Point.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6198I *fsswtr_id printer_id* **DEVICE BACK-SPACED**

Explanation: The data set at the Operator Orientation Point has been backspaced.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6199I *fsswtr_id printer_id* **DEVICE FORWARD-SPACED**

Explanation: The data set at the Operator Orientation Point has been forward-spaced.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6200I *fsswtr_id printer_id* **DEVICE SYNCH, ORDSYR1-4=synch_flags**

Explanation: This message is a general DEVICE SYNCH message. It displays the flags from the JES-to-FSA SYNCH ORDER.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPFA

EZY6201E *aaa bbbbbbbb* **ABENDED**, *cccccc* **CODE=dddd** **REASON=eeeeeeee**

Explanation: This message is presented by the NPF JES Capture Point ESTAE exit. The message contains information from an ABEND that occurred as follows:

aaa Identifies the component that ABENDED. The value is either FSS or FSA.

bbbbbbbb
 If *aaa* is FSS, *bbbbbbbb* is the FSS writer ID. If *aaa* is FSA, *bbbbbbbb* is the printer ID.

cccccc indicates either a SYSTEM or a USER ABEND.

dddd contains the ABEND code.

eeeeeeee contains the reason code.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPFE

EZY6202E EZAPPF E FSS/FSA SVC DUMP COMPLETE

Explanation: The SVC DUMP is complete. This message is issued from the NPF JES Capture Point ESTAE exit in conjunction with IEA794I.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPF E

EZY6203E FSS/FSA SVC DUMP FAILED, R15=aaaa, REASON=bbbb

Explanation: The SVC DUMP failed. The return code from SDUMP (from register 15) is output in field *aaaa*. The reason code is output in field *bbbb*.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPF E

EZY6204E LRB FAILING INSTR WAS aaaaaaaaaa

Explanation: This message displays the Last-RB failing instruction in field *aaaaaaaaaa*.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPF E

EZY6205E LRB PSW = aaaaaaaa aaaaaaaa ILC = b IC = cc

Explanation: This message displays the Last-RB PSW in field *aaaaaaaa aaaaaaaa*, the Instruction Length Code in *b*, and the Interrupt Code in field *cc*.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPF E

EZY6206E LRB aaa = register register register register

Explanation: This message displays four Last-RB registers starting with the register identified in field *aaa*. *aaa* is one of the following: R0, R4, R8, or RC.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPF E

EZY6207R REPLY "P" WHEN EZAPPF E MAY PERCOLATE, OR "TOE" (TIME-OF-ERROR) OR "LRB" (LAST-RB) FOR ERROR INFO RE-DISPLAY

Explanation: This message enables you to display diagnostic data before the FSA or FSS task abends. EZAPPF E provides a basic ESTAE snapshot for these tasks. The operator must respond with either of the following replies:

P Causes EZAPPFE to percolate, which ends the FSA or FSS task, and takes an SVCDUMP.

TOE Redispays the instruction at the time of the error, PSW, and registers, which can be used for diagnostic purposes.

LRB Redispays the LAST RB information from the FSA or FSS task, which can be used for diagnostic purposes.

System Action: The FSA or FSS task will end and an SVCDUMP will be taken.

User or Operator Response: Save the system log and the dump for problem determination.

System Programmer Response: Check system log and console for related error messages.

Source Data Set: EZAPPFE

Procedure Name: SFSEEP1

EZY6214E TOE FAILING INSTR WAS *aaaaaaaaaaaa*

Explanation: This message displays the Time-Of-Error failing instruction in field *aaaaaaaaaaaa*.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPFE

EZY6215E TOE PSW = *aaaaaaaa aaaaaaaa* ILC = *b* IC = *cc*

Explanation: This message displays the Time-Of-Error PSW in field *aaaaaaaa aaaaaaaa*, the Instruction Length Code in *b*, and the Interrupt Code in field *cc*.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPFE

EZY6216E LRB *aaa* = *register register register register*

Explanation: This message displays four Last-RB registers starting with the register identified in field *aaa*. *aaa* is one of the following: R0, R4, R8, or RC.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Use the information in this message as an aid to problem analysis.

Source Data Set: EZAPPFE

EZY6221E *fsswtr_id printer_id* FSA-CONNECT FSIREQ FAILED, SSOBRETN=*fsi_return_code*

Explanation: The FSA-CONNECT FSIREQ was unsuccessful. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6222E • EZY6227E

EZY6222E *fsswtr_id printer_id* **FSA-CONNECT FSIREQ FAILED, R15=register_15_contents**

Explanation: The FSA-CONNECT FSIREQ was unsuccessful. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6223E *fsswtr_id printer_id* **INVALID UCB ADDRESS PASSED FROM JES**

Explanation: The FSA detected that an invalid UCB address was passed from JES. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6224E *fsswtr_id printer_id* **GETMAIN FOR SMF WORK AREA FAILED**

Explanation: The FSA tried to issue GETMAIN for its SMF work area. The GETMAIN failed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

Source Data Set: EZAPPFA

EZY6225E *fsswtr_id printer_id* **GETMAIN FOR PIPELINE WORK AREAS FAILED**

Explanation: The FSA tried to issue GETMAIN for its PIPELINE work area. The GETMAIN failed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

Source Data Set: EZAPPFA

EZY6226E *fsswtr_id printer_id* **GETMAIN FOR FSI CKPT WORK AREA FAILED**

Explanation: The FSA tried to issue GETMAIN for its FSI checkpoint work area. The GETMAIN failed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

Source Data Set: EZAPPFA

EZY6227E *fsswtr_id printer_id* **GETMAIN FOR SJF WORK AREA FAILED**

Explanation: The FSA tried to issue GETMAIN for its Scheduler JCL Facility work area. The GETMAIN failed.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

System Action: Processing continues.

Source Data Set: EZAPPFA

EZY6228E GETMAIN FOR DEVICE DRIVER SDCB FAILED

Explanation: The FSA tried to issue GETMAIN for its Device Driver's Control Block. The GETMAIN failed.

System Action: The FSA ends.

User or Operator Response: None.

System Programmer Response: Review any accompanying MVS system messages to try to determine if the GETMAIN failure was due to a correctable error. If the error cannot be corrected, contact IBM Support.

Source Data Set: EZAPPFA

EZY6230E *fsswtr_id printer_id* FSA-DISCONNECT FSIREQ SSOBRETN=*fsireq_return_code*

Explanation: The FSA did not DISCONNECT successfully. The return code from FSIREQ is listed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6231E *fsswtr_id printer_id* FSA-DISCONNECT FSIREQ R15=*fsi_return_code*

Explanation: The FSA did not DISCONNECT successfully. The return code from the Subsystem Interface is listed. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6232E *fsswtr_id printer_id* ABNORMAL FSA DISCONNECT REQUESTED

Explanation: The FSA cannot DISCONNECT successfully. The time frame for disconnection is invalid. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6233E *fsswtr_id printer_id* FSA DISCONNECT REFUSED, DEVICE STILL UP

Explanation: The FSA cannot DISCONNECT successfully. The device is still active. This is an internal error. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6234E • EZY6239E

EZY6234E *fsswtr_id printer_id* **START DEVICE ORDER REFUSED, ALREADY UP**

Explanation: The FSA has received an asynchronous ORDER request, via an FSIREQ, requesting that the FSA start its device. The FSA is already started. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6235E *fsswtr_id printer_id* **STOP DEVICE ORDER REFUSED, DEVICE NOT UP**

Explanation: The FSA has received an asynchronous ORDER request, via an FSIREQ, requesting that the FSA stop its device. The FSA is not started. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6236E *fsswtr_id printer_id* **UNSUPPORTED FSA ORDER, ORDID=*fsa_order_id***

Explanation: The FSA has received an ORDER that is unsupported. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6237E *fsswtr_id printer_id* **GETDS FSIREQ FAILED, R15=*fsireq_return_code***

Explanation: A GETDS FSIREQ call failed. The return code from the call is listed in the message. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6238E *fsswtr_id printer_id* **GETREC FSIREQ FAILED, R15=*fsireq_return_code***

Explanation: A GETREC FSIREQ call failed. The return code from the call is listed in the message. This is an internal error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6239E *fsswtr_id printer_id* **FREEREC FSIREQ FAILED, R15=*fsireq_return_code***

Explanation: A FREEREC FSIREQ call failed. The return code from the call is listed in the message. This is an internal error.

System Action: Data set processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6240E *fsswtr_id printer_id* **CKPT FSIREQ FAILED, R15=fsireq_return_code**

Explanation: A CKPT FSIREQ call failed. The return code from the call is listed in the message. This is an internal error.

System Action: Data set processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6241E *fsswtr_id printer_id* **GETREC I/O ERROR (GLRIOE)**

Explanation: An I/O error occurred during GETREC processing.

System Action: The data set is held and processing continues.

User or Operator Response: None.

System Programmer Response: Analyze accompanying MVS system messages to determine the cause of the error. Contact IBM Support if you are unable to correct the error.

Source Data Set: EZAPPFA

EZY6242E *fsswtr_id printer_id* **INVALID FSI PARMLIST IN GETREC (GLRIPL)**

Explanation: During a GETREC FSIREQ, an invalid FSI parmlist was encountered. This is an internal error.

System Action: Data set Processing ends.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6243E *fsswtr_id printer_id* **GETREC PROCESSING ERROR (GLRLGE)**

Explanation: During a GETREC FSIREQ, an internal logic error was encountered.

System Action: The dataset is held, and processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6244E *fsswtr_id printer_id* **DEVICE DRIVER EXIT ID/RC = exit_vector_offset/return_code**

Explanation: A nonzero return code was received from the device driver exit.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6251W *fsswtr_id printer_id* **FSA REQUESTED TERMINATION (FIT SEND)**

Explanation: An FSI SEND was issued to JES to request self-initiated termination.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPFA

EZY6252W *fsswtr_id printer_id* **SJF ERROR, SJFREQ R15=sjfreq_return_code**

Explanation: An SJFREQ call failed, with the return code listed in the message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFA

EZY6253W *fsswtr_id printer_id* **SJF RETRIEVE ERROR, SJFREREAS=sjfreq_reason_code**

Explanation: An SJF RETRIEVE error was detected that is not critical.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: On return from the SJF request, it was determined that some keywords were not found. This is assumed to be because the system JDTs are at a lower level than the list of keywords in the FSA module. Since that list is ordered by release date of keywords, it can be assumed that the needed information has been found.

Source Data Set: EZAPPPFA

EZY6255W *fsswtr_id printer_id* **RELDS FSIREQ ERROR, RC=relds_return_code**

Explanation: A RELDS FSIREQ call was issued for the data set at the stacker. The call was unsuccessful.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFA

EZY6256W *fsswtr_id printer_id* **TRACKING FSI SEND ERROR, RC=return_code**

Explanation: A tracking SEND FSIREQ call was issued for the data set at the Operator Orientation Point. The call failed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPPFA

EZY6257W *fsswtr_id printer_id* **FSI SEND FSIREQ ERROR, RC=fsi_req_return_code**

Explanation: An FSISEND call was issued to JES, but was unsuccessful.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6258W *fsswtr_id printer_id* **SJF UNKNOWN ERROR, SJFREAREAS=sjfreq_reason_code**

Explanation: An SJFREQ call was unsuccessful. The reason could not be determined.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Contact IBM Support.

Source Data Set: EZAPPFA

EZY6259W *fsswtr_id printer_id* **NOTIFY SSI ERROR, R15=ssi_return_code, USER=user_list**

Explanation: An attempt was made to notify a list of users that data set processing has been completed. The attempt was unsuccessful.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If you are unable to correct the conditions responsible for the NOTIFY failure, contact IBM Support.

Source Data Set: EZAPPFA

EZY6260W *fsswtr_id printer_id* **NOTIFY ERROR, SSOBRETN=ssob_return_code**

Explanation: An attempt was made to notify a list of users that data set processing has been completed. The attempt was unsuccessful.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If you are unable to correct the conditions responsible for the NOTIFY failure, contact IBM Support.

Source Data Set: EZAPPFA

EZY6261W *fsswtr_id printer_id* **NOTIFY ERROR, SSNUERCD=ssnu_error_code**

Explanation: An attempt was made to notify a list of users that data set processing has been completed. The attempt was unsuccessful.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If you are unable to correct the conditions responsible for the NOTIFY failure, contact the IBM Software Support Center to report this error.

Source Data Set: EZAPPFA

Chapter 8. EZYFxxxx messages

EZYFS01I SECURE_CTRLCONN value must be CLEAR, SAFE, or PRIVATE

Explanation: While processing the FTP.DATA file, the server or client encountered the SECURE_CTRLCONN statement with a parameter value that was not CLEAR, SAFE, or PRIVATE. The only valid values are CLEAR, SAFE, and PRIVATE.

System Action: The line containing the error is ignored. FTP.DATA file processing continues with the next line of 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 statement. See the *z/OS Communications Server: IP Configuration Guide* for information about configuring the FTP server to use the TLS security mechanism.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS02I SECURE_DATACONN value must be CLEAR, SAFE, PRIVATE, or NEVER

Explanation: While processing the FTP.DATA file, the server or client encountered the SECURE_DATACONN statement with a parameter value that was not CLEAR, SAFE, PRIVATE, or NEVER. The only valid values are CLEAR, SAFE, PRIVATE, and NEVER.

System Action: The line containing the error is ignored. FTP.DATA file processing continues with the next line of 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 statement. See the *z/OS Communications Server: IP Configuration Guide* for information about configuring the FTP server to use the TLS security mechanism.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS03I SECURE_MECHANISM value must be GSSAPI, or TLS

Explanation: While processing the FTP.DATA file, the client encountered the SECURE_MECHANISM statement with a parameter value that was not GSSAPI or TLS. The only valid values are GSSAPI or TLS.

System Action: The line containing the error is ignored. FTP.DATA file processing continues with the next line of 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 statement. See the *z/OS Communications Server: IP Configuration Guide* for information about configuring the FTP server to use the TLS security mechanism.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS04I Maximum allowable number of ciphersuite statements has been specified

Explanation: While processing the FTP.DATA file, the server or client encountered the CIPHERSUITE statement after the maximum number of valid CIPHERSUITE statements have been processed. Only 20 CIPHERSUITE statements can be processed.

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

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

System Programmer Response: Delete the extra ciphersuite statements.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS05I Ciphersuite name is not supported

Explanation: While processing the FTP.DATA file, the server or client encountered the CIPHERSUITE statement with a parameter value that was not recognized.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line of 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 statement. See the *z/OS Communications Server: IP Configuration Guide* for information about configuring the FTP server to use the TLS security mechanism.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS06I Ciphersuite name has already been specified

Explanation: While processing the FTP.DATA file, the server or client encountered the CIPHERSUITE statement with a parameter name that has already been specified.

System Action: The line containing the error is ignored. Processing of the FTP.DATA file continues with the next line of 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 no duplicate ciphersuite names.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS07I *smfkeyword* value must be TYPE119 or a blank

Explanation: While processing the FTP.DATA file, the server encountered the SMF keyword parameter with a parameter value that was not TYPE119 or a blank. The only valid values for the *smfkeyword* parameter are TYPE119 and a blank.

smfkeyword is the FTP.DATA keyword for SMF record.

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

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

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

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata()

EZYFS08I Some characters cannot be translated between *codeset_1* and *codeset_2*

Explanation: The SBDATACONN statement specified the codesets *codeset_1* and *codeset_2*. The iconv() function was used to build a table for each of the 256 single byte character codepoints. Some of the codepoints do not have an equivalent codepoint.

System Action: The tables are built and will be used during data transfer by FTP. If FTP detects a data byte during the transfer of the data that cannot be translated using the tables, FTP will use substitution characters if substitution is enabled, or otherwise fail the transfer.

User or Operator Response: None.

System Programmer Response: Specify SBDATACONN with codeset names that convert all of the codepoints. However, you can use the table if your data does not contain any of the untranslatable characters. To see which of the codepoints cannot be translated, start the client or server with the following trace specified in the FTP.DATA file:

DEBUG UTL ; utility services trace

Source Data Set: EZAFTPNX, EZAFTPCK

Procedure Name: setup_translate_tables, locsite

EZYFS09I SBSUBCHAR must be either SPACE or a hexadecimal character.

Explanation: While processing the FTP.DATA file, the server encountered the SBSUBCHAR statement with a parameter value that was not SPACE, or a hexadecimal character. The only valid values for the SBSUBCHAR parameter are SPACE or a hexadecimal character.

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. See the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS10I ENCODING value must be either SBCS or MBCS

Explanation: The ENCODING statement in the FTP.DATA file has a value that is not SBCS or MBCS. The only valid values are SBCS and MBCS.

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 statement. See the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS11I Multi-byte encoding does not support *codepage* as a file system codepage

Explanation: The format of the MBDATACONN statement in the FTP.DATA file is the following:

MBDATACONN (file_system_cp,network_transfer_cp).

The multi-byte encoding support for FTP allows codepages IBM-1388 and UTF-8 to be specified as the file system codepage.

codepage is the codepage name that was entered as the file system codepage.

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 use one of the supported file system codepages. See the *z/OS Communications Server: IP Configuration Reference* for information about the MBDATACONN statement.

Source Data Set: EZAFTPEP

Procedure Name: verifyMBdataconn

EZYFS12I Multi-byte encoding does not support *codepage* as a network transfer codepage

Explanation: The format of the MBDATACONN statement in the FTP.DATA file is the following:

MBDATACONN (file_system_cp,network_transfer_cp).

The multi-byte encoding support for FTP allows codepage IBM-5488 to be specified as the network transfer codepage.

codepage is the codepage name that was entered as the network transfer codepage.

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 use one of the supported network transfer codepages. See the *z/OS Communications Server: IP Configuration Reference* for information about the MBDATACONN statement.

Source Data Set: EZAFTPEP

Procedure Name: verifyMBdataconn

EZYFS13I No conversion available to *cp_name1* from *cp_name2*

Explanation: The MBDATACONN statement in the FTP.DATA file specified codepage names, but there is no supported code set converter for the code sets (codepages) that are specified.

cp_name1 is the codepage name **to** which the code is converted.

cp_name2 is the codepage name **from** which the code is converted.

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 use one of the supported pairs of code sets (codepages). See the *z/OS XL C/C++ Programming Guide* for information about supported code set converters and valid code set names.

Source Data Set: EZAFTPEJ

Procedure Name: verifyMBdataconn

EZYFS16I SECURE_PASSWORD reset to REQUIRED

Explanation: After processing the FTP.DATA statements, the server cross-checked the values of the SECURE_LOGIN and SECURE_PASSWORD statements and found them to be in a combination that is not valid. SECURE_PASSWORD is coded with a value of OPTIONAL and SECURE_LOGIN has the value NO_CLIENT_AUTH. NO_CLIENT_AUTH indicates that the server does not request a certificate from the client, but SECURE_PASSWORD OPTIONAL means that a certificate is required for a session protected by the TLS security mechanism.

System Action: Because the server will not request a certificate from the client, SECURE_PASSWORD is reset to REQUIRED. As a result, a password will be required for authentication of a TLS secured login.

User or Operator Response: If the new value is acceptable, no action is required. Otherwise, contact the system programmer to change the SECURE_LOGIN setting to a value other than NO_CLIENT_AUTH in order to permit SECURE_PASSWORD OPTIONAL.

System Programmer Response: If necessary, update the FTP.DATA file with the correct values for SECURE_LOGIN and SECURE_PASSWORD. Refer to *z/OS Communications Server: IP Configuration Reference* for information about the statements of the FTP.DATA file.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFS20I Binary tagged file translated with current data connection translation table

Explanation: The file that is to be transferred was tagged binary but the data type is ASCII. This is a warning to the user that the file will be translated.

System Action: The file transfer continues.

User or Operator Response: If it is acceptable that the translation was done in ASCII, then no action is necessary. However, if no translation should have been done, change the transfer type to binary and transfer the file again. See the *z/OS UNIX System Services Command Reference* for information about the CHTAG command. See the *z/OS Communications Server: IP User's Guide and Commands* for information about the FTP TYPE subcommand.

System Programmer Response: None

Source Data Set: EZAFTPSM

Procedure Name: hfs_sndFile()

EZYFS50I ID=sessionID CONN starts Client IPaddr=ipaddr hostname=hostname

Explanation: This log entry is made by the FTP daemon when it accepts a client connection request. The keyword CONN identifies this entry as a connection log entry.

sessionID uniquely identifies the FTP session between a client and a server. The identifier is created by combining the jobname of the FTP daemon with a 5-digit number in the range 00000–99999. This identifier is in each log entry for the session until message EZYFS52I, which is the last entry for the session.

ipaddr is the IP address of the FTP client. The IP address may be either an IPv4 or an IPv6 address.

hostname is the name of the FTP client. If the name cannot be resolved, UNKNOWN is displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logCONN

EZYFS51I ID=*sessionID* CONN fails Reason=*reason* Text=*text*

Explanation: This log entry is made by the FTP daemon or the server when a connection request fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

reason is a number code for the failure.

text is an explanation of the failure.

System Action: The FTP connection ends.

User or Operator Response: None.

System Programmer Response: Use the reason code in the log entry to choose from the following responses:

- 1 Examine your FTCHKIP user exit to determine why the connection was rejected.
- 2 If the FTP server trace was not active with the INT option, use the MODIFY operator command to activate the trace with the INT option. Ask the client to connect again to the server. Contact the IBM support center and provide the log entry message number, the reason number of the failure, and the FTP server trace of the failure.
- 3 If the FTP server trace was not active with the INT option, use the MODIFY operator command to activate the trace with the INT option. Ask the client to connect again to the server. Contact the IBM support center and provide the log entry message number, the reason number of the failure, and the FTP server trace of the failure.
- 4 If the FTP server trace was not active with the INT option, use the MODIFY operator command to activate the trace with the INT option. Ask the client to connect again to the server. Contact the IBM support center and provide the log entry message number, the reason number of the failure, and the FTP server trace of the failure.
- 5 If the FTP server trace was not active with the INT option, use the MODIFY operator command to activate the trace with the INT option. Ask the client to connect again to the server. Contact the IBM support center and provide the log entry message number, the reason number of the failure, and the FTP server trace of the failure.
- 6 If the FTP server trace was not active with the INT option, use the MODIFY operator command to activate the trace with the INT option. Ask the client to connect again to the server. Contact the IBM support center and provide the log entry message number, the reason number of the failure, and the FTP server trace of the failure.
- 7 Restart the daemon with a larger region size and ask the client to connect again to the server.
- 8 Restart the daemon with a larger region size and ask the client to connect again to the server.
- 9 Restart the daemon with a larger region size and ask the client to connect again to the server.
- 10 Contact the IBM support center and provide the log entry message number and reason number of the failure.

Source Data Set: EZAFPBU

Procedure Name: logCONN

EZYFS52I ID=*sessionID* CONN ends Input=*bytesIn* Output=*bytesOut*

Explanation: This log entry is made by the FTP server when an FTP session with the client ends. The keyword CONN identifies this entry as a connection log entry.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

bytesIn is the count, in bytes, of the data that was transferred into the server with the data transfer commands (STOR, STOU, and APPE).

bytesOut is the count, in bytes, of the data that was transferred from the server with the data transfer command (RETR) and the list commands (LIST and NLST).

If a count is greater than or equal to one gigabyte (that is, 1,073,741,824 bytes), the count is displayed as a number of gigabytes rounded to the nearest hundredth of a gigabyte. Some examples are:

- 1,073,741,824 bytes is displayed as 1.00 GB
- 2,147,483,648 bytes is displayed as 2.00 GB
- 2,415,919,104 bytes is displayed as 2.25 GB

System Action: The FTP connection ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logCONN

EZYFS54I ID=*sessionID* SECURE OK Mechanism=*mechanism*

Explanation: This log entry is made by the FTP server to indicate that the connection with the client is protected by a security mechanism.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

mechanism is either TLS, TLS-P, or GSSAPI.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logSECURE

EZYFS55I ID=*sessionID* SECURE fails Reason=*reason* Text=*text*

Explanation: This log entry is made by the FTP server when a request for security protection fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

reason is a number code for the failure

text is an explanation of the failure.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Use the reason code number in the message to determine your action as follows:

- 1 If your installation wants to provide TLS security for FTP connections, add the EXTENSIONS AUTH_TLS and the KEYRING statements to your FTP.DATA file and restart the FTP daemon. Otherwise, use the message EZYFS50I for this session ID to determine the identity of the FTP client that is requesting a secure session and handle appropriately for your installation.
- 2 If your installation wants to allow private data connections with TLS secure connections, change the SECURE_DATACONN statement in your FTP.DATA file to CLEAR or PRIVATE and restart the FTP daemon. Otherwise, use the message EZYFS50I for this session ID to determine the identity of the FTP client that is requesting a secure session with private data connections and handle appropriately for your installation.
- 3 If your installation wants to provide Kerberos security for FTP connections, add the EXTENSIONS AUTH_GSSAPI statement to your FTP.DATA file and restart the FTP daemon. See the *z/OS Communications Server: IP Configuration Guide* for more information. Otherwise, use the message EZYFS50I for this session ID to determine the identity of the FTP client that is requesting a secure session and handle appropriately for your installation.
- 4 Use the MODIFY operator command to set the FTP server trace as follows: MODIFY jobname,DEBUG=(SOC(3),SEC). Then ask the client to try to connect to the FTP server again. Contact the IBM support center and provide the server trace and the log entry message.
- 5 If your installation wants to allow the USER command before the session is protected by a security mechanism, add the SECURE_FTP ALLOWED statement to the FTP.DATA file and restart the daemon. If you require the security protection, then the client must send an AUTH command before the USER command is sent.

EZYFS56I • EZYFS57I

- 6 If your installation wants to allow the PASS command before the session is protected by a security mechanism, add the SECURE_FTP ALLOWED statement to the FTP.DATA file and restart the daemon. If you require the security protection, then the client must send an AUTH command before the PASS command is sent.
- 7 If your installation wants to allow commands before the session is protected by a security mechanism, add the SECURE_FTP ALLOWED statement to the FTP.DATA file and restart the daemon. If you require the security protection, then the client must send an AUTH command as the first command.
- 8 If GSSAPI connections are required, provide an FTP server that supports the IPv4 transport.

Source Data Set: EZAFTPBU

Procedure Name: logSECURE

EZYFS56I ID=*sessionID* ACCESS OK USERID=*userid*

Explanation: This log entry is made by the FTP server to indicate a successful verification of the user that logged in with the USER command. If the client changes the user ID during the session with another USER command, another ACCESS entry is made to record the new user ID.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

userid is the name that was entered on the USER command

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logACCESS

EZYFS57I ID=*sessionID* ACCESS fails USERID=*userid* Reason=*reason_code* Text=*text*

Explanation: This log entry is made by the FTP server when the access verification fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

userid is the login name (user ID).

reason_code is the code that corresponds to the failure.

text is an explanation of the failure.

System Action: FTP continues. The client is still in session but is not logged in.

User or Operator Response: None.

System Programmer Response: Use the *reason_code*, described in the following table, to determine your response. If the identity of the client is needed, use the log entry message EZYFS50I for this session ID to determine the identity of the FTP client that is accessing the FTP server with the USER or PASS command. If an FTP server trace is needed, ensure that the ACC trace option is activated for the server trace before the client logs in. Use the MODIFY operator command or an FTP.DATA file statement to activate the trace.

reason_code	System programmer response
1	Ask the client to reissue the USER and PASS commands, using the correct format of the old_password/new_password/new_password on the PASS command.
2	Ask the client to reissue the USER and PASS commands, using the same password for both occurrences of new password .
3	The FTCHKPWD user exit of your installation rejected the user ID. Update the exit to accept the user ID.
4	Ask the client to reissue the USER and PASS commands, using the correct password for the user ID.

reason_code	System programmer response
5	Ask the client to reissue the USER and PASS commands with valid user ID and password values.
6	Ensure that programs being loaded from the address space of the FTP server are defined as program controlled. A corresponding ICH420I message is issued to identify the uncontrolled library where the load was done.
7	Ask the client to reissue the USER and PASS commands, using the format of old_password/new_password/new_password on the PASS command to provide a new password for the user ID.
8	Ask the client to reissue the USER and PASS commands, using the format of old_password/new_password/new_password on the PASS command with a new password that meets the installation's requirements for passwords.
9	Determine why the user ID access is revoked and restore its access, or ask the client to reissue the USER and PASS commands with a user ID whose access has not been revoked.
10	Determine why the user ID is no longer known, or ask the client to reissue the USER and PASS commands with a user ID that is known to the system.
11	Access to the user database by user ID is a function provided by the getpwnam() function of the operating system. Contact the IBM support center and provide the log entry message number, reason number of the failure, and the FTP server trace for the failure.
12	The process group ID is set by the setgid() function of the operating system. If the FTP server ACC trace was active when the setgid() failed, the server made a trace entry with the errno of the failure. The errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in <i>z/OS UNIX System Services Messages and Codes</i> . Contact the IBM support center and provide the log entry message number, reason number of the failure, and the FTP server trace for the failure.
13	Contact the IBM support center and provide the log entry message number, reason number of the failure, and the FTP server trace for the failure.
14	Contact the IBM support center and provide the log entry message number, reason number of the failure, and the FTP server trace for the failure.
15	Contact the IBM support center and provide the log entry message number, reason number of the failure, and the FTP server trace for the failure.
16	Contact the IBM support center and provide the log entry message number, reason number of the failure, and the FTP server trace for the failure.
17	<p>Your installation has defined in the security product (for example, RACF) a SERVAUTH class and a profile for the port as follows: EZB.FTP.systemname.ftpdemonname.PORTxxxx where xxxx is the port number of the FTP daemon. The user login is denied because of the following:</p> <ul style="list-style-type: none"> • The session is protected by TLS. • The FTP.DATA file has the statement SECURE_LOGIN VERIFY_USER. • The user ID does not have read access to the profile. <p>Give the user ID read access to the profile if the user needs to use TLS security.</p>
18	The connection is protected by TLS security. However, the client did not send a certificate to the FTP server during the TLS handshake process. Either the client must send a certificate or you should change the FTP.DATA file statement to SECURE_LOGIN NO_CLIENT_AUTH.
19	The USER command is disabled for this client. Inform the client that he must end his current FTP session and connect again to provide the USER command and the PASS command with the correct password.
20	To allow anonymous logins with an active security mechanism, code the ANONYMOUSLEVEL statement in FTP.DATA file with a value greater than 2.
21	Determine whether the certificate or ticket used to authenticate the user is associated in the security product with a different user.

EZYFS58I • EZYFS60I

Source Data Set: EZAFTPBU

Procedure Name: logACCESS

EZYFS58I ID=*sessionID* ALLOC permission denied MVS DSN=*dsname*

Explanation: This log entry is made by the FTP server when it determines through the use of a SAF-compliant security product (for example, RACF) that the end user does not have permission to access the MVS data set for the command that is being processed.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

dsname is the full MVS name of the data set.

System Action: The MVS data set is not allocated.

User or Operator Response: None.

System Programmer Response: If the user should be allowed to access the data set, update the security product to give permission for access to the MVS data set to the user identified in the log entry message EZYFS56I. Otherwise, use the message EZYFS50I for this session ID to determine the identity of the FTP client that is requesting access to the data set and handle appropriately for your installation.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS59I ID=*sessionID* ALLOC permission denied HFS filename=*filename*

Explanation: This log entry is made by the FTP server when it determines that the end user does not have permission to access the HFS file for the command that is being processed.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

filename is the full HFS file name of the file.

System Action: The HFS file is not allocated.

User or Operator Response: None.

System Programmer Response: If the user should be allowed to access the file, update the permissions of the file and any directories in the path to allow access to the HFS file for the user identified in the log entry message EZYFS56I. Otherwise, use the message EZYFS50I for this session ID to determine the identity of the FTP client that is requesting access to the file and handle appropriately for your installation.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS60I ID=*sessionID* ALLOC OK *action* MVS DSN=*dsname*

Explanation: This log entry is made by the FTP server when it successfully allocates an MVS data set.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates whether a new data set was created during allocation or an existing data set was used. The values for *action* are the following:

Create A new data set was created.

Use An existing data set was used.

dsname is the full MVS name of the data set.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS61I ID=*sessionID* ALLOC DDNAME=*ddname* VOLSER=*volser* DSORG=*dsorg* DISP=(*disp*)

Explanation: This log entry provides additional information about an MVS data set allocation.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

ddname is the ddname associated with the allocation.

volser is the first volume serial number associated with the allocation.

dsorg is the data set organization of the data set. Examples are DSORG=PS (physical sequential) and DSORG=PO (partitioned).

disp is the status and disposition of the allocation. Examples are DISP=(NEW,CATLG,CATLG) when a new data set is created and DISP=(SHR,KEEP) when an existing data set is retrieved.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS62I ID=*sessionID* ALLOC SMS Storclas=*storclass* Mgmtclas=*mgmtclass* Dataclas=*dataclass*

Explanation: This log entry provides additional information about an allocation for a new MVS data set.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

storclass is the storage class of a new SMS-managed data set.

mgmtclass is the management class of a new SMS-managed data set.

dataclass is the data class of a new SMS-managed data set.

System Action: FTP continues.

User or Operator Response: None

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS63I ID=*sessionID* ALLOC fails *action* MVS DSN=*dsname*

Explanation: This log entry is made by the FTP server when it attempts to allocate an MVS data set and the allocation fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates whether a new data set was to be created during allocation or an existing data set was to be used. The values for action are the following:

Create Create a new data set.

Use Use an existing data set.

dsname is the full MVS name of the data set.

EZYFS64I • EZYFS66I

System Action: The MVS data set is not allocated.

User or Operator Response: None.

System Programmer Response: Look for log entry messages EZYFS64I and EZYFS65I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS64I ID=*sessionID* ALLOC SVC 99 RC=*rc* ERROR=*err* INFO=*info*

Explanation: This log entry provides additional information about an unsuccessful allocation for an MVS data set. The information is returned by SVC 99 and is defined in the dynamic allocation chapters of the *z/OS MVS Programming: Authorized Assembler Services Guide*.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

rc is the DYNALLOC Return Code when the allocation failed.

err is the hexadecimal DYNALLOC Error Reason Code returned when the allocation failed.

info is the hexadecimal DYNALLOC Information Reason Code returned when the allocation failed.

System Action: The MVS data set is not allocated.

User or Operator Response: None.

System Programmer Response: See the Error Reason Codes table in the *z/OS MVS Programming: Authorized Assembler Services Guide* for the meaning and action to be taken for the error. Also, look for the log entry message EZYFS65I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS65I ID=*sessionID* ALLOC Message=*text*

Explanation: This log entry provides additional information about an unsuccessful allocation of an MVS data set.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

text is the text of an extracted message from dynamic allocation. The following are examples of the failure messages that are extracted when the FTP server attempts to write to a data set that is currently being edited:

EZYFS65I FTP ID=FTPD100001 ALLOC Message=IKJ56225I DATA SET USER33.TEST.S.ALLOC.A1
ALREADY IN USE, TRY LATER+

EZYFS65I FTP ID=FTPD100001 ALLOC Message=IKJ56225I DATA SET IS ALLOCATED TO ANOTHER JOB OR USER

The IKJ56225I message numbers are data management numbers associated with dynamic allocation.

System Action: The MVS data is not allocated.

User or Operator Response: None.

System Programmer Response: Look at the log entry for message EZYFS64I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS66I ID=*sessionID* ALLOC SVC 99 S99ERSN = *reason*

Explanation: This log entry provides additional information about an unsuccessful allocation for an MVS data set. The information is returned by SVC 99 and is defined in the dynamic allocation chapters of the *z/OS MVS Programming: Authorized Assembler Services Guide*.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

reason is the hexadecimal SMS error reason code S99ERSN.

System Action: The MVS data set is not allocated.

User or Operator Response: Contact the system programmer.

System Programmer Response: Refer to *z/OS MVS Programming: Authorized Assembler Services Guide* for the SMS error reason codes and their meanings. Also, look for the log entry messages EZYFS64I and EZYFS65I for additional information.

Source Data Set: EZAFTPGU

Procedure Name: logALLOC

EZYFS67I ID=*sessionID* ALLOC OK *action* HFS filename=*filename*

Explanation: This log entry is made by the FTP server when it successfully allocates an HFS file.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates whether a new file was created during allocation or an existing file was used. The values for *action* are the following:

Create A new file was created.

Use An existing file was used.

filename is the full HFS file name.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS68I ID=*sessionID* ALLOC fails *action* HFS filename=*filename*

Explanation: This log entry is made by the FTP server when it attempts to allocate an HFS file and the allocation fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates whether a new file was to be created during allocation or an existing file was to be used. The values for *action* are the following:

Create Create a new file.

Use Use an existing file.

filename is the full HFS file name.

System Action: The HFS file is not allocated.

User or Operator Response: None.

System Programmer Response: Look for log entry message EZYFS69I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS69I ID=*sessionID* ALLOC Errno=*err* Text=*text*

Explanation: This log entry provides additional information about an unsuccessful allocation for an HFS file.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

err is the errno value that is returned from the fopen() function that attempted the allocation.

text is a description of the error.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: See the *z/OS XL C/C++ Run-Time Library Reference* for a description of the failure errno values for the fopen() function. *errnos* are the UNIX System Services Return Codes. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*. If you are not able to correct the problem using the failure descriptions, contact the IBM support center and provide the log entry message number and *errno* value of the failure.

Source Data Set: EZAFTPBU

Procedure Name: logALLOC

EZYFS70I ID=*sessionID* DEALL OK *action* MVS DSN=*dsname*

Explanation: This log entry is made by the FTP server when it successfully deallocates an MVS data set.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates the action taken when the data set was deallocated. The following are the values for action:

Release Deallocate and keep the data set.

Delete Deallocate and delete the data set.

A data set is usually released at the end of a transfer command. The exception is when a newly created data set is deleted because the transfer failed and the CONDDISP DELETE option was chosen in the FTP.DATA file or by the SITE command. An MVS data set is also deallocated with delete when the DELE command is processed by the server.

dsname is the full MVS name of the data set.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS71I ID=*sessionID* DEALL fails *action* MVS DSN=*dsname*

Explanation: This log entry is made by the FTP server when it attempts to deallocate an MVS data set and the deallocation fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates the action taken when the data set was deallocated. The following are the values for action:

Release Deallocate and keep the data set.

Delete Deallocate and delete the data set.

A data set is usually released at the end of a transfer command. The exception is when a newly created data set is deleted because the transfer failed and the CONDDISP DELETE option was chosen in the FTP.DATA file or by the SITE command. An MVS data set is also deallocated with delete when the DELE command is processed by the server.

dsname is the full MVS name of the data set.

System Action: The MVS data set is not deallocated.

User or Operator Response: None.

System Programmer Response: Look for log entry message EZYFS72I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS72I ID=*sessionID* DEALL SVC 99 RC=*rc* ERROR=*err* INFO=*info*

Explanation: This log entry provides additional information about an unsuccessful deallocation for an MVS data set. The information is returned by SVC 99 and is defined in the dynamic allocation chapters of the *z/OS MVS Programming: Authorized Assembler Services Guide*.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

rc is the decimal DYNALLOC Return Code when the deallocation failed.

err is the decimal DYNALLOC Error Reason Code returned when the deallocation failed.

info is the hexadecimal DYNALLOC Information Reason Code returned when the deallocation failed.

System Action: The MVS data set is not deallocated.

User or Operator Response: None.

System Programmer Response: See the Error Reason Codes table in the *z/OS MVS Programming: Authorized Assembler Services Guide* for meaning and action to be taken for the error.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS73I ID=*sessionID* DEALL SVC 99 S99ERSN = *reason*

Explanation: This log entry provides additional information about an unsuccessful deallocation for an MVS data set. The information is returned by SVC 99 and is defined in the dynamic allocation chapters of the *z/OS MVS Programming: Authorized Assembler Services Guide*.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

reason is the hexadecimal SMS error reason code S99ERSN.

System Action: The MVS data set might not be deallocated correctly.

User or Operator Response: Contact the system programmer.

System Programmer Response: Refer to *z/OS MVS Programming: Authorized Assembler Services Guide* for the SMS error reason codes and their meanings. Also, look for the log entry message EZYFS72I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS74I ID=*sessionID* ALLOC SVC 99 S99INFO = *info*

Explanation: This log entry provides additional information about a successful allocation for an MVS data set. The information is returned by SVC 99 and is defined in the dynamic allocation chapters of the *z/OS MVS Programming: Authorized Assembler Services Guide*.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

info is the hexadecimal informational reason code S99INFO.

System Action: The MVS data set is allocated. FTP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: The S99INFO information reason code provides additional information about any errors that occurred during the allocation that did not cause the allocation to fail. Refer to the *z/OS MVS Programming: Authorized Assembler Services Guide* for a description of the S99INFO value.

Source Data Set: EZAFTPGU

Procedure Name: logALLOC

EZYFS75I ID=*sessionID* DEALL SVC 99 S99INFO = *info*

Explanation: This log entry provides additional information about a successful deallocation for an MVS data set. The information is returned by SVC 99 and is defined in the dynamic allocation chapters of the *z/OS MVS Programming: Authorized Assembler Services Guide*.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

info is the hexadecimal informational reason code S99INFO.

System Action: The MVS data set is deallocated. FTP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: The S99INFO information reason code provides additional information about any errors that occurred during the deallocation that did not cause the deallocation to fail. Refer to the *z/OS MVS Programming: Authorized Assembler Services Guide* for a description of the S99INFO value.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS77I ID=*sessionID* DEALL OK *action* HFS *filename*=*filename*

Explanation: This log entry is made by the FTP server when it successfully deallocates an HFS file.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates the action taken when the file was deallocated. The following are the values for action:

Release Deallocate and keep the file.

Delete Deallocate and delete the file.

A file is usually released at the end of a transfer command. The exception is when a newly created file is deleted because the transfer failed and the CONDDISP DELETE option was chosen in the FTP.DATA file or by the SITE command. A file is also deallocated and deleted when the DELE command is processed by the server.

filename is the full HFS file name.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS78I ID=*sessionID* DEALL fails action HFS filename=*filename*

Explanation: This log entry is made by the FTP server when it attempts to deallocate an HFS file and the deallocation fails.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action indicates the action taken when the file was deallocated. The following are the values for action:

Release Deallocate and keep the file.

Delete Deallocate and delete the file.

A file is usually released at the end of a transfer command. The exception is when a newly created file is deleted because the transfer failed and the CONDDISP DELETE option was chosen in the FTP.DATA file or by the SITE command. A file is also deallocated and deleted when the DELE command is processed by the server.

filename is the full HFS file name.

System Action: The file is not deallocated.

User or Operator Response: None.

System Programmer Response: Look for log entry message EZYFS79I for additional information.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS79I ID=*sessionID* DEALL Errno=*err* Text=*text*

Explanation: This log entry provides additional information about an unsuccessful deallocation for an HFS file.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

err is the errno value that is returned by the function that was used to deallocate the file.

text is a description of the error.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: Look for message EZYFS78I that precedes this message in the log. If the action in the message is Release, the function `fclose()` was used to deallocate the file. If the action in the message is Delete, the function `remove()` was used to deallocate the file. The *z/OS XL C/C++ Run-Time Library Reference* describes the failure values for the `fclose()` and `remove()` functions. If you are not able to correct the problem using the failure descriptions, contact the IBM Software Support Center and provide the log entry message number, the *errno* value, and the description of the failure.

Source Data Set: EZAFTPBU

Procedure Name: logDEALL

EZYFS80I ID=*sessionID* TRANS Reply=*reply*

Explanation: This log entry is the last entry made by the FTP server for a data transfer command process. It logs the reply that is sent to the client to report the completion of the data transfer.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

reply is data that begins with a 3-digit reply code followed by the text for the reply. The first digit of the 3-digit reply indicates success or failure defined as follows:

EZYFS81I • EZYFS83I

- 2 Positive Completion reply. The requested action has been successfully completed.
- 4 Transient Negative Completion reply. The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again.
- 5 Permanent Negative Completion reply. The command was not accepted and the requested action did not take place.

For more information about FTP reply codes, see RFC 959. See Appendix B, “Related protocol specifications (RFCs),” on page 555 for information about accessing RFCs.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logTRANS

EZYFS81I ID=*sessionID* TRANS MVS DSN=*dsname*

Explanation: This log entry is the first entry made by the FTP server for a data transfer command process for an MVS data set. It logs the name of the MVS data set that the server is sending from or receiving into.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

dsname is the full MVS name of the data set.

System Action: FTP continues.

User or Operator Response: None

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logTRANS

EZYFS82I ID=*sessionID* TRANS HFS filename=*filename*

Explanation: This log entry is the first entry made by the FTP server for a data transfer command process for an HFS file. It logs the name of the HFS file that the server is sending from or receiving into.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

filename is the full HFS file name.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logTRANS

EZYFS83I ID=*sessionID* TRANS Stru=*stru* Mode=*mode* Type=*type* Input=*bytesIn*

Explanation: This log entry contains information about an input data transfer from the perspective of the FTP server. Processing for a STOR, STOU, or APPE command results in an input data transfer.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

stru is the data structure for the transfer. The following are the values for *stru*:

F file structure

R record structure

mode is the transmission mode. The following are the values for *mode*:

S stream mode

B block mode

C compressed mode

type is the data type. The following are the values for *type*:

A ASCII

E EBCDIC

I binary (image)

U UCS2

B DBCS

bytesIn is the number of bytes transferred on the data connection for this command. If a number is greater than or equal to one gigabyte (that is, 1 073 741 824 bytes), the number is displayed as a number of gigabytes rounded to the nearest hundredth of a gigabyte. Some examples are:

1 073 741 824 bytes is displayed as 1.00 GB

2 147 483 648 bytes is displayed as 2.00 GB

2 415 919 104 bytes is displayed as 2.25 GB

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logTRANS

EZYFS84I ID=*sessionID* TRANS Stru=*stru* Mode=*mode* Type=*type* Output=*bytesOut*

Explanation: This log entry contains information about an output data transfer from the perspective of the FTP server. Processing for a RETR command results in an output data transfer.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

stru is the data structure for the transfer. The following are the values for *stru*:

F file structure

R record structure

mode is the transmission mode. The following are the values for *mode*:

S stream mode

B block mode

C compressed mode

type is the data type. The following are the values for *type*:

A ASCII

E EBCDIC

I binary (image)

U UCS2

EZYFS85I • EZYFS86I

B DBCS

bytesOut is the number of bytes transferred on the data connection for this command. If a number is greater than or equal to one gigabyte (that is, 1 073 741 824 bytes), the number is displayed as a number of gigabytes rounded to the nearest hundredth of a gigabyte. Some examples are:

- 1 073 741 824 bytes is displayed as 1.00 GB
- 2 147 483 648 bytes is displayed as 2.00 GB
- 2 415 919 104 bytes is displayed as 2.25 GB

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logTRANS

EZYFS85I ID=*sessionID* TRANS JES JobID=*jobid* DDNAME=*ddname* SpoolFile=*spool*

Explanation: This log entry is the first entry made by the FTP server for a data transfer that returns the output from a job when the FTP filetype is JES.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

jobid is the job identifier associated with the JES job.

ddname is the ddname associated with spool file.

spool is the name of the spool file.

This message is followed in the log by a message EZYFS84I and EZYFS80I.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logTRANS

EZYFS86I ID=*sessionID* TRANS Confidence=*confidence_level*

Explanation: This log entry is written by the FTP server to report the confidence level in the successful completion of a file transfer when CHKConfidence TRUE is specified or defaulted to in the server FTP.DATA data set. The preceding message EZYFS81I or EZYFS82I contains the name of the data set or file being stored by the server.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

confidence_level is the level of confidence that the FTP server has in the completion of the transfer. Values for *confidence_level* are:

High While successful completion of the transfer cannot be guaranteed, the FTP server did not detect a reason to doubt completion of the transfer.

NoEOF The FTP server detected a missing EOF marker in a STRU R or MODE B or C inbound file. This level is reported only when no other problem is detected.

Low The FTP server detected a problem with completion of the transfer such as the failure of the client to respond after the transfer or another reported error. Low overrides NoEOF if both conditions are present.

Unknown

This confidence level is reported when the direction of the file transfer was outbound. If an error occurs

| while shutting down the data connection, then confidence in an outbound transfer will be reported as **Low**.
 | In all other cases, the confidence level is reported as **Unknown** because the server cannot perform any other
 | checks on an outbound transfer.

| **System Action:** FTP continues.

| **User or Operator Response:** Check to see whether the transfer completed successfully and retry, if necessary.

| **System Programmer Response:** None.

| **Source Data Set:** EZAFTPBU

| **Procedure Name:** logTRANS()

EZYFS91I ID=*sessionID* SUBMIT JES JobID=*jobid*

Explanation: This log entry is made by the FTP server for a data transfer that submitted a job when the FTP filetype is JES.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

jobid is the job identifier associated with the JES job. The job identifier is used later by the client end user to retrieve the output of the submitted job.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logSUBMIT

EZYFS92I ID=*sessionID* QUERY SQL filename=*filename*

Explanation: This log entry is the first entry made for a data transfer of a report requested when the FTP filetype is SQL.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

filename is the full HFS file name or MVS data set name that contains the select statement used to make the query.

This message is followed in the log by messages EZYFS84I and EZYFS80I.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPBU

Procedure Name: logQUERY

EZYFS95I ID=*sessionID* ABEND *action* COMP_CODE=*code* DSN=*dsname*

Explanation: This log entry records the system completion code when an ABEND occurs during an event for an MVS data set.

sessionID uniquely identifies the FTP session between a client and a server. Message EZYFS50I is the first message in the log for this session ID.

action is the name of the operation in progress when the failure occurred. The values are READ, WRITE, and CLOSE.

code is the ABEND completion code.

dsname is the full MVS name of the data set.

EZYFT01I • EZYFT05I

The following is an example of an ABEND message that occurs when the receiving data set is too small to hold the data that is transferred:

EZYFS95I ID=FTPDJG100001 ABEND WRITE COMP_CODE=D37-04 DSN=USER33.T00BIG

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: See the *z/OS MVS System Codes* for the system programmer response for the completion code in this message.

Source Data Set: EZAFTPBU

Procedure Name: logABEND

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

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPDM

EZYFT02E Filename exceeds maximum valid length of *max*.

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

System Action: FTP continues.

User or Operator Response: None.

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

Source Data Set: EZAFTPEP

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

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

System Action: The statement is ignored. Processing continues.

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

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

System Action: Processing continues.

Source Data Set: EZAFTPEP

EZYFT05I *Volume_serial_list* is not a valid VOLUME parameter.

Explanation: While processing the VOLUME statement in the FTP.DATA file, a syntax error was detected in *volume_serial_list*.

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

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

System Programmer Response: Correct the FTP.DATA file to contain the correct value for *volume_serial_list*. See the *z/OS Communications Server: IP Configuration Reference* for more information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata()

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: If an external reply catalog is desired, either place a copy of ftpdrrply.cat in the 'C' subdirectory of your nlspath, or use the REPLYLANGUAGE parameter in the FTP.DATA file to establish one or more reply catalogs to be used. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

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

System Programmer Response: Correct the value specified for UMASK in the FTP.DATA file. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

User or Operator Response: None.

System Programmer Response: Verify the search path to the ETC.SERVICES file and verify that an entry exists in the appropriate ETC.SERVICES file for FTP. If the PORT start option is specified, or if PORT 21 is an acceptable default port, this message may be ignored. Otherwise, add an entry for the ftp server to the appropriate etc.services file. Refer to the *z/OS Communications Server: IP Configuration Reference* information about configuring the etc.services file for the FTP server.

Source Data Set: EZAFTPDM

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

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

System Action: Processing continues.

EZYFT12E • EZYFT16E

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM

EZYFT12E **socket error : error**

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPSK

EZYFT13E **bind error : error**

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

System Action: The FTP server is ended.

User or Operator Response: None.

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

Source Data Set: EZAFTPDS

EZYFT14E **listen error : error**

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

System Action: The FTP server is ended.

User or Operator Response: None.

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

Source Data Set: EZAFTPSK

EZYFT15E **selectex error : error**

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

System Action: The FTP server is ended.

User or Operator Response: None.

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

Source Data Set: EZAFTPSK

EZYFT16E **accept error : error**

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

System Action: The FTP server is ended.

User or Operator Response: None.

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

Source Data Set: EZAFTPSK

EZYFT17E getsockname error : *error*

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

System Action: The FTP server is ended.

User or Operator Response: None.

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

Source Data Set: EZAFTPSK

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

Explanation: The messages issued by FTP server or client is retrieved from the message catalog in *catalog*.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM and EZAFTPCY

EZYFT19E Unrecognized code page name: '*name*'

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

System Action: The statement is ignored. Processing continues.

User or Operator Response: None.

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

Source Data Set: EZAFTPEP

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

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

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

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

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

Source Data Set: EZAFTPEP

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

EZYFT22E • EZYFT24E

Source Data Set: EZAFTPEP

EZYFT22E Unexpected end of file.

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPDY

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

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

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

User or Operator Response: None.

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

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

Source Data Set: EZAFTPNX

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

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

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

User or Operator Response: None.

System Programmer Response: Valid code sets are listed in the *z/OS XL C/C++ Programming Guide* in the section about the code set converters supplied. If the requested code sets are double-byte code sets, the use of CTRLCONN or SBADATACONN to establish translate tables is not supported, and the statement should be removed from the FTP.DATA file. FTP validates the results of ICONV and returns an error if there are missing code points. If the code sets are valid single-byte code sets, an internal error has occurred. Contact the IBM Software Support Center with this message and the FTP trace output, if it is available.

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

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

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

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

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

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

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

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

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

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

Source Data Set: EZAFTPNX

EZYFT34W Sigaction for *signal* failed : error (*errnoerrnojr*)**

Explanation: The FTP server encountered an error while attempting to setup the signal handler for the signal specified by *signal*. If the signal handler is not correctly enabled, the server will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

SIGABND	handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information about the abend will be lost and certain resources might not be properly cleaned up.
SIGCHLD	handler controls cleanup of zombie processes when a client connection is ended. If sigaction fails for SIGCHLD, zombie processes will not be cleaned up when a client connection is ended.
SIGTERM	handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, the FTP server will not be cleanly terminated when an MVS operator STOP command is issued or when the server process is killed.
SIGURG	handler controls the processing of Out of Band data, such as the ABOR subcommand. If sigaction fails for SIGURG, OOB data will not be received or processed by the server.

error is the error returned by the C run-time library for the failing sigaction() call.

errno is the UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDM, EZAFTPRX, EZAFTPHC

EZYFT40E FTP server initialization failed - error

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

System Action: The FTP server is ended.

User or Operator Response: None.

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

Source Data Set: EZAFTPDM

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPSK

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

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPNX

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

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPDM

EZYFT44E Translate table is too small.

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPDY

EZYFT45E Translate table is too large.

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

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

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

User or Operator Response: None.

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

Source Data Set: EZAFTPDY

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

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

statement. *line_number* is the number of the line in the FTP.DATA file which contains the error. *column_number* is the approximate location of the error within the line.

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

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

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

Source Data Set: EZAFTPEP

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

Explanation: While processing the FTP.DATA file, the FTP server encountered a keyword that was valid for another FTP server or the FTP client, but that is unsupported by this server. Or, the FTP server is ignoring *keyword* because it is inconsistent with another keyword in FTP.DATA. These keywords will be ignored when the default or explicit ANONYMOUSLEVEL value does not support the keyword:

ANONYMOUSHFSFILEMODE
ANONYMOUSHFSDIRMODE
ANONYMOUSFILETYPEJES
ANONYMOUSFILETYPESEQ
ANONYMOUSFILETYPESQL
EMAILADDRCHECK
ANONYMOUSFILEACCESS

These keywords will be ignored when ANONYMOUS is not explicitly defined in FTP.DATA.

ANONYMOUSHFSFILEMODE
ANONYMOUSHFSDIRMODE
ANONYMOUSFILETYPEJES
ANONYMOUSFILETYPESEQ
ANONYMOUSFILETYPESQL
EMAILADDRCHECK
ANONYMOUSFILEACCESS
ANONYMOUSLEVEL

System Action: The FTP server ignores the keyword.

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

System Programmer Response: If the FTP.DATA file is used only by the FTP server, 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 about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

User or Operator Response: Contact the system programmer.

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

Source Data Set: EZAFTPDM

EZYFT49I Statement *statement* parameter must be one of the following: *list_of_supported_parameters*

Explanation: A statement in FTP.DATA is coded with a parameter that is not supported.

statement is the statement that is coded incorrectly.

list_of_supported_parameters is a list of parameters that are supported for this statement.

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: Change statement in FTP.DATA file to specify a supported parameter. Refer to *z/OS Communications Server: IP Configuration Reference* for information about supported parameters for statements coded in FTP.DATA.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata()

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

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

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

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

System Programmer Response: Correct the value of the JESRECFM parameter in the FTP.DATA file to be a valid record format. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

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

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

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

User or Operator Response: Contact the system programmer.

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

Source Data Set: EZAFTPSK

EZYFT54E FTP initialization terminated because TCP/IP Base feature of OS/390 is not enabled.

Explanation: The FTP server 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 server will only run if `TCP/IP_BASE` is enabled.

System Action: The FTP initialization is ended.

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 daemon. If `TCP/IP_BASE` is not part of your system there is no action to take and this FTP server cannot be used.

Source Data Set: EZAFTPDM

EZYFT55E STARTDIrectory value must be either MVS or HFS.

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

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

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

System Programmer Response: Correct the value of the STARTDIrectory parameter in the FTP.DATA file to be a valid column heading. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZYFT56W Unable to register with WLM, diagnostic code = *reason*

Explanation: While processing the FTP.DATA file, the keyword WLMClustername was encountered. Processing this keyword has failed.

| *reason* is the decimal reason code returned from the WLM sysplex registration service IWMSRSRG. See *z/OS MVS Programming: Workload Management Services* for an explanation of *reason*.

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

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

| **System Programmer Response:** Use the explanation of *reason* to correct the error.

EZYFT57I FTP registering with WLM as group = *group* host = *host*

Explanation: FTP has registered with WLM to create a new name in the DNS database. The group name in this message can now be used by FTP clients to connect to any in an equivalent group of FTP servers running in a sysplex. This is done to balance connections.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZYFT58W SMF STD overrides *smfxxxx* on line *line*

Explanation: While processing the FTP.DATA file, the FTP Server encountered both SMF STD and SMFxxxx. If SMF STD is coded, none of the SMFxxxx options can be coded.

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

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

System Programmer Response: If using SMF STD, remove all SMFxxxx options from FTP.DATA. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

EZYFT59I FTP shutdown complete.

Explanation: The FTP daemon has been terminated either by an MVS operator STOP command, or by an OMVS kill command. No new FTP sessions will be accepted.

System Action: The FTP daemon ends. Any FTP sessions currently active are not effected.

User or Operator Response: None.

System Programmer Response: None.

EZYFT60I • EZYFT63E

Source Data Set: EZAFTPDH

EZYFT60I Both CCXLATE and CTRLCONN were specified. CCXLATE will be ignored.

Explanation: The FTP daemon has encountered both CCXLATE and CTRLCONN keywords in FTP.DATA. Each is used to specify an initial translate table for the control connection, but CTRLCONN is preferred. When both keywords are present in FTP.DATA, the CTRLCONN value will be used for the server configuration and the CCXLATE value will be ignored.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: To avoid this message, remove either the CCXLATE or CTRLCONN statement from the FTP.DATA file.

Source Data Set: EZAFTPPEP

EZYFT61I Both XLATE and SBADATACONN were specified. XLATE will be ignored.

Explanation: The FTP daemon has encountered both XLATE and SBADATACONN keywords in FTP.DATA. Each is used to specify the initial translate table for the data connection, but SBADATACONN is preferred. When both keywords are present in FTP.DATA, the XLATE value will be ignored. The SBADATACONN value will be used for the server configuration unless DD:SYSFTSX has been defined.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: To avoid this message, remove either the XLATE or SBADATACONN statement from the FTP.DATA file.

Source Data Set: EZAFTPPEP

EZYFT62E Unable to use file '*file*' for translate tables for the control connection.

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

System Action: The CCXLATE statement is ignored. The FTP Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, default 7-bit translate tables will be built.

User or Operator Response: None.

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

Source Data Set: EZAFTPNX

EZYFT63E Unable to use CCXLATE configuration parameter *name*

Explanation: The FTP server was unable either to locate or to use the file determined by the CCXLATE statement in the FTP.DATA file. (The file name is *hlq.name.TCPXLBIN* unless this name was superseded by an environment variable called *_FTPXLATE_name* that defines a file name.) This message will be preceded or followed by another message with additional information. If FTP tracing was enabled during initialization, additional messages will precede this one with more specific detail about the error encountered. Possible errors include an invalid header record or incorrect file length. (The file must contain exactly 768 bytes of data.)

System Action: The CCXLATE statement is ignored. The FTP Server will continue through the search order of translation tables attempting to find a good translation table. If no translation tables are found, default 7-bit translate tables will be built.

User or Operator Response: None.

System Programmer Response: If file specified by the CCXLATE statement is desired, determine why it was not

usable, correct the problem, and stop and restart the FTP server.

Source Data Set: EZAFTPNX

EZYFT64E Unable to use XLATE configuration parameter *name*

Explanation: The FTP server was unable either to locate or to use the file determined by the XLATE statement in the FTP.DATA file. (The file name is *hlq.name.TCPXLBIN* unless this name was superseded by an environment variable called `_FTPXLATE_name` that defines a file name.) This message will be preceded or followed by another message with additional information. If FTP tracing was enabled during initialization, additional messages will precede this one with more specific detail about the error encountered. Possible errors include an invalid header record or incorrect file length. (The file must contain exactly 768 bytes of data.)

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

User or Operator Response: None.

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

Source Data Set: EZAFTPNX

EZYFT65E Filename defined by environment variable *variable* exceeds the maximum length of *length*

Explanation: The FTP.DATA file contains a 'CCLXATE *name*' (or 'XLATE *name*') statement to specify translate tables for the control (or data) connection, but the `_FTPXLATE_name` environment variable defines a file name that is too long and cannot be used for the translate tables file. This message will be followed by another message that displays the CCXLATE (or XLATE) parameter from FTP.DATA.

System Action: The CCXLATE (or XLATE) statement is ignored. The FTP Server will continue through the search order of translation tables attempting to find a good translation table.

User or Operator Response: None.

System Programmer Response: If tracing was active when the FTP server was initialized, all existing environment variables will be displayed in the trace. Ensure that the `_FTPXLATE_name` variable is correct and that it specifies a file name that is not longer than *length* characters.

Source Data Set: EZAFTPNX

EZYFT67E UCOUNT value *value* is not valid. Value must be P, or an integer from 1 through 59.

Explanation: While processing the FTP.DATA file, the server encountered the UCOUNT parameter with a value that was not a numeric value from 1 through 59, nor was it the letter P. Only values 1 to 59, or the letter P, are valid values for the UCOUNT parameter.

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

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

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

Source Data Set: EZAFTPPEP

Procedure Name: read_ftpdata()

EZYFT68I ANONYMOUSFILEACCESS value must be MVS, HFS, or BOTH

Explanation: While processing the FTP.DATA file, the ANONYMOUSFILEACCESS parameter was encountered with a value other than MVS, HFS, or BOTH. These are the only valid values for ANONYMOUSFILEACCESS.

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

EZYFT69I • EZYFT71E

User or Operator Response: Contact the System programmer with the error message to have the statement in 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 about the parameters of the FTP.DATA file.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata()

Procedure Name: read_ftpdata()

EZYFT69I *parameter value value is not valid -- parameter value must be three octal digits*

Explanation: While processing the FTP.DATA file, a parameter was encountered that requires a 3-digit octal value. The actual value specified, *value*, is not three octal digits. *Parameter* is the parameter in error.

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

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

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

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata()

EZYFT70I SERVER-FTP: ACCEPTING CONNECTIONS

Explanation: The FTP Server is accepting connections from all known common INET (CINET) stacks. This message appears when the MVS operator has just started a CINET stack, and the FTP server has recognized the new stack as well as other CINET stacks active at the time.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDS

EZYFT71E *operation failed in function - error*

Explanation: The FTP server encountered a severe error.

operation is the operation in progress.

function is the C Library function that failed and returned an error.

error is the C run-time library error message for the failure. For more information, refer to *z/OS Language Environment Run-Time Messages*.

System Action: The FTP server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error indicated by *error*. If you are unable to resolve the problem, contact the IBM Software Support Center and report the contents of this message.

Source Data Set: N/A

Procedure Name: N/A

EZYFT72I FTP SERVER SUSPENDING INCOMING CONNECTIONS

Explanation: The FTP server stopped accepting incoming connections because a common INET (CINET) stack just started. While the FTP server is reinitializing, no connections can be accepted. This condition is temporary. Message EZYFT70I will follow when the FTP server is accepting connections again.

System Action: Processing continues.

User or Operator Response: Wait for message EZYFT70I before attempting to connect to the FTP server.

System Programmer Response: None.

Source Data Set: EZAFTPSD

EZYFT73I ANONYMOUS LOGINS NOT ALLOWED -- REASON IS *reason code*

Explanation: The FTP server found the ANONYMOUS statement in FTP.DATA; however, the server is not allowing anonymous logins due to an FTP configuration error. The *reason code* indicates the nature of the configuration error. Furthermore, the FTP trace will contain messages further describing the error. When reason code is:

- 1 The FTP daemon was unable to locate the anonymous root directory. A possible reason for this is the userid specified on the ANONYMOUS statement is not defined in the user database.
- 2 The FTP daemon was unable to locate the executable file *ls* in the bin subdirectory of the anonymous root directory. A possible reason for this is the system programmer did not set up the root directory as directed in the *z/OS Communications Server: IP User's Guide and Commands*.
- 3 The FTP daemon was unable to locate the executable file *sh* in the bin subdirectory of the anonymous root directory. A possible reason for this is the system programmer did not set up the root directory as directed in the *z/OS Communications Server: IP User's Guide and Commands*.
- 4 The FTP daemon was unable to locate the executable file *ftpdns* in the /usr/sbin/ subdirectory of the anonymous root directory. A possible reason for this is the system programmer did not set up the root directory as directed in the *z/OS Communications Server: IP User's Guide and Commands*.
- 5 ANONYMOUSLEVEL value must be greater than or equal to 3 to support the SURROGATE parameter as an anonymous password.

System Action: FTP logs the reason it is not allowing anonymous logins. FTP continues processing. Attempts to log in as userid **anonymous** will fail.

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

System Programmer Response: If you do not want the FTP server to accept anonymous logins, remove the ANONYMOUS statement from FTP.DATA. If you do want the FTP server to accept anonymous logins, correct the error indicated by *reason code*. Inspecting the FTP trace for error messages related to not allowing anonymous logins may provide useful information. Trace must be active at startup to capture error messages related to anonymous FTP configuration. Refer to *z/OS Communications Server: IP User's Guide and Commands* for information about configuring the FTP server for anonymous logins.

Source Data Set: EZAFTPDM

EZYFT74I Using internal translate tables for the control connection.

Explanation: A CTRLCONN statement in the FTP.DATA file specified a value of FTP_STANDARD_TABLE. FTP will use its internal translate tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), for the control connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

EZYFT75I Using internal translate tables for the data connection.

Explanation: An SBDATACONN statement in the FTP.DATA file specified a value of FTP_STANDARD_TABLE. FTP will use its internal translate tables, which are the same as the tables that are shipped in TCPXLBIN(STANDARD), for the data connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPNX

EZYFT76I EXTensions value *value* not recognized.

Explanation: While processing the FTP.DATA file, the server encountered the EXTensions statement with a value that was not SIZE, MDTM, REST_STREAM, AUTH_GSSAPI, AUTH_TLS, or UTF8. Only these values are valid for the EXTensions parameter.

value is the value encountered.

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

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

System Programmer Response: Correct the FTP.DATA file to contain a valid value for the specified parameter. See the *z/OS Communications Server: IP Configuration Reference* for information about the statements of the FTP.DATA file.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpData

EZYFT77W Unable to set address space nonswappable - *error*

Explanation: The FTP daemon was unable to set its address space to nonswappable as requested in the FTP.DATA file.

error is the error message returned by the C run-time library for the failing __mlockall() routine.

System Action: Processing continues. The FTP daemon will run with swappable memory.

User or Operator Response: None.

System Programmer Response: Determine the meaning of the error message using *z/OS Language Environment Run-Time Messages*. If the text of the message is EDC5139I Operation not permitted then verify that the daemon (FTPD) has at least READ access to the FACILITY class resource BPX.STOR.SWAP. Refer to *z/OS Communications Server: IP Configuration Reference* for information about configuring the FTP daemon to be nonswappable.

Source Data Set: EZAFTPDM

Procedure Name: set_nonswap

EZYFT78I lowport value *lowport* cannot exceed highport value *highport*

Explanation: FTP encountered an error while processing a PASSIVEDATAPORTS statement in the FTP.DATA file. The value specified for the lowest allowed port number is greater than the value specified for the highest allowed port number. This is not allowed.

lowport is the value specified for the lowest allowed port number.

highport is the value specified for the highest allowed port number.

System Action: The current statement is ignored. FTP continues processing.

User or Operator Response: Correct the erroneous statement in the FTP.DATA file, and restart FTP. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about statements in the FTP.DATA file.

System Programmer Response: None.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata()

EZYFT80I MODIFY COMMAND IGNORED - NO VALUE WAS SPECIFIED FOR *trace*

Explanation: A MODIFY command was issued to the FTP server to change the values for the FTP general trace or for the FTP extended trace. No trace value was specified. A command to change a trace must have at least one value specified.

trace is the trace that is being changed and can be one of the following:

DEBUG

FTP general trace

DUMP FTP extended trace

System Action: The command is ignored.

User or Operator Response: Verify that the MODIFY command was entered correctly. See the *z/OS Communications Server: IP Diagnosis Guide* for information about diagnosing FTP server problems with traces and for the values of the DEBUG and DUMP parameters of the MODIFY command.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

Procedure Name: mvs_command_handler

EZYFT81I UNSUPPORTED *trace* **VALUES -** *value_1 ... value_n*

Explanation: A MODIFY command was issued to the FTP server to change the values for the FTP general trace or for the FTP extended trace.

trace is the trace that is being changed and can be one of the following:

DEBUG

FTP general trace

DUMP FTP extended trace

The list shows the values that were specified that are not supported.

System Action: The unsupported values are ignored.

User or Operator Response: Verify that the MODIFY command was entered correctly. See the *z/OS Communications Server: IP Diagnosis Guide* for information about diagnosing FTP server problems with traces and for the values of the DUMP parameter of the MODIFY command that apply to JES.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

Procedure Name: mvs_command_handler

EZYFT82I ACTIVE SERVER TRACES - *name_1 ... name_n*

Explanation: This message displays the active FTP server general traces. This message is issued when a MODIFY command is issued to the FTP server to change the values for the FTP general trace.

name_x can be one of the following:

ACC access control (logging in)

CMD command trace

INT program initialization and termination

FLO function flow

FSC(1) file services -- level 1

FSC(2) file services -- level 2

EZYFT83I • EZYFT84I

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
JES JES processing
NONE no trace is active
PAR parser details
SEC security functions
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
UTL utility functions

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

Procedure Name: mvsv_command_handler

EZYFT83I ACTIVE SERVER DUMPIDS - *id_1 ... id_n*

Explanation: A MODIFY command was issued to the FTP server to change the values for the FTP extended trace. This message displays the active FTP server extended trace IDs (DUMPIDS). If no extended trace IDs are active, then the word **NONE**. is displayed.

System Action: FTP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

Procedure Name: mvsv_command_handler

EZYFT84I *parameter* OBSOLETE - USE DUMP TO CONTROL FTP EXTENDED TRACING FOR JES

Explanation: A MODIFY command was issued to the FTP server to enable or disable the FTP extended trace for JES processing.

parameter is one of the following:

- JDUMP — enable extended tracing for JES
- NOJDUMP — disable extended tracing for JES

System Action: The MODIFY command is ignored.

User or Operator Response: Refer to the *z/OS Communications Server: IP Diagnosis Guide* for information about diagnosing FTP server problems with Traces and for the values of the DUMP parameter of the MODIFY command that apply to JES.

System Programmer Response: None.

Source Data Set: EZAFTPDF

Procedure Name: mvs_command_handler

EZYFT85I PASSIVEDATACONN value must be UNRESTRICTED or NOREDIRECT

Explanation: While processing the FTP.DATA file, the server encountered the PASSIVEDATACONN parameter with a parameter value that was not UNRESTRICTED or NOREDIRECT. The only valid values for the PASSIVEDATACONN parameter are NORESTRICTED and NOREDIRECT.

System Action: The line containing the error is ignored. The default of UNRESTRICTED will be used. 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. For more information, refer to *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT86I USER TRACE IS OBSOLETE - *parameter* IS IGNORED

Explanation: A MODIFY command was issued to the FTP server to activate the user trace. The user trace is obsolete and is replaced by the user ID filter for the FTP server general and extended traces.

parameter is one of the following:

UTRACE

Start user trace

NOUTRACE

Stop user trace

System Action: The MODIFY command is ignored.

User or Operator Response: See the *z/OS Communications Server: IP Diagnosis Guide* for information about diagnosing FTP Server problems with traces and for the syntax of the user ID filter for the FTP server general and extended traces.

System Programmer Response: None.

Source Data Set: EZAFTPDF

Procedure Name: mvs_command_handler

EZYFT87I VALUE NOT SUPPORTED FOR *trace* STATEMENT

Explanation: A trace statement in FTP.DATA specified a value that is not supported by FTP.

trace is one of the following:

DEBUG

FTP general trace

DUMP

FTP extended trace

System Action: The statement is ignored.

User or Operator Response: See the *z/OS Communications Server: IP Configuration Reference* for more information about the DEBUG and DUMP statements and FTP.DATA data set statements. See the *z/OS Communications Server: IP*

EZYFT89I • EZYFT92I

User's Guide and Commands for information about changing local site defaults using FTP.DATA for the FTP client.

System Programmer Response: None.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT89I USERID FILTER - *filter*

Explanation: A MODIFY command was issued to the FTP server to change values for the FTP general trace (DEBUG) or extended trace (DUMP).

filter is the name of the filter for the userids for which tracing is active.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

Procedure Name: mvs_command_handler

EZYFT90I IPADDR FILTER - *filter*

Explanation: A MODIFY command was issued to the FTP server to change values for the FTP general trace (DEBUG) or extended trace (DUMP).

filter is the name of the filter for the IP addresses for which tracing is active.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAFTPDPF

Procedure Name: mvs_command_handler

EZYFT91I PORTCOMMAND value must be ACCEPT or REJECT

Explanation: While processing the FTP.DATA file, the server encountered the PORTCOMMAND parameter with a parameter value that was not ACCEPT or REJECT. The only valid values for the PORTCOMMAND parameter are ACCEPT and REJECT.

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. See the *z/OS Communications Server: IP Configuration Reference* for more information about the PORTCOMMAND parameter.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT92I PORTCOMMANDPORT value must be UNRESTRICTED or NOLOWPORTS

Explanation: While processing the FTP.DATA file, the server encountered the PORTCOMMANDPORT parameter with a parameter value that was not UNRESTRICTED or NOLOWPORTS. The only valid values for the PORTCOMMANDPORT parameter are UNRESTRICTED and NOLOWPORTS.

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. See the *z/OS Communications Server: IP Configuration Reference* for more information about the PORTCOMMANDPORT parameter.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT93I PORTCOMMANDIPADDR value must be UNRESTRICTED or NOREDIRECT

Explanation: While processing the FTP.DATA file, the server encountered the PORTCOMMANDIPADDR parameter with a parameter value that was not UNRESTRICTED or NOREDIRECT. The only valid values for the PORTCOMMANDIPADDR parameter are NORESTRICTED and NOREDIRECT.

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. See the *z/OS Communications Server: IP Configuration Reference* for more information about the PORTCOMMANDIPADDR parameter.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT94I ANONYMOUSLEVEL value must be greater than or equal to 3 to support the SURROGATE parameter as an anonymous password

Explanation: In the server FTP.DATA, SURROGATE is defined as the password for anonymous users, and ANONYMOUSLEVEL is less than 3. In order to use SURROGATE as the password for anonymous users, ANONYMOUSLEVEL must be greater than or equal to 3.

System Action: The line containing the error is ignored. Anonymous logins will be disabled. FTP continues.

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

System Programmer Response: Set the ANONYMOUSLEVEL to 3 or greater. See the *z/OS Communications Server: IP Configuration Reference* for more information about ANONYMOUSLEVEL.

Source Data Set: EZAFTPDM

Procedure Name: check_surrogate_support

EZYFT95I Server setup for TLS failed

Explanation: The FTP server detected an error condition when it was processing a request for TLS authentication from an FTP client. The error condition prevented the server from completing the request for a TLS protected session.

System Action: The request for authentication failed. If a secure FTP session is required by the server, the session is active but no commands are accepted until an AUTH command is successfully processed. If a secure FTP session is not required, the session setup continues but the session is not TLS protected.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine the cause of the error by examining the FTP server trace. The SEC trace type should be active before the client attempts to connect to the FTP server. The trace entry that precedes message EZYFT95I in the trace describes the reason for the failure.

The error message is displayed when one of the following has occurred:

EZYFT96I • EZYFT98I

- No keyring is defined in the FTP.DATA file for the server or a keyring file name is specified but there is no associated stash file. Refer to *z/OS Communications Server: IP Configuration Reference* for more information about KEYRING.
- One of the TLS setup functions used by the FTP server failed. Refer to *z/OS Cryptographic Service System Secure Sockets Layer Programming* to determine how to obtain diagnostic information for the TLS programming functions.

Source Data Set: EZAFTPFR, EZAFTPRX

Procedure Name: auth, main

EZYFT96I TLS handshake failed

Explanation: The FTP server detected an error condition when it was processing a request for TLS authentication from an FTP client. The error condition occurred during the server to client handshake process and prevented the server from completing the request for a TLS protected session.

System Action: The request for authentication failed. If a secure FTP session is required by the server, the session is active but no commands are accepted until an AUTH command is successfully processed. If a secure session is not required, the session setup continues but the session is not TLS protected.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine the cause of the error by examining the FTP server trace. The SEC trace type and the SOC level 3 trace type must be active before the client attempts to connect to the FTP server. The trace entries that precede this message in this trace describe the reason for the failure.

Source Data Set: EZAFTPFR, EZAFTPRX

Procedure Name: auth, main

EZYFT97I SECURE_FTP value must be REQUIRED or ALLOWED

Explanation: While processing the FTP.DATA file, the server or client encountered the SECURE_FTP statement with a parameter value that was not REQUIRED or ALLOWED. The only valid values are REQUIRED and ALLOWED.

System Action: The line containing the error is ignored. FTP.DATA file processing continues with the next line of 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 statement. See the *z/OS Communications Server: IP Configuration Guide* for information about configuring the FTP server to use the TLS security mechanism.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT98I SECURE_LOGIN value must be NO_CLIENT_AUTH, REQUIRED, or VERIFY_USER

Explanation: While processing the FTP.DATA file, the server encountered the SECURE_LOGIN statement with a parameter value that was not NO_CLIENT_AUTH, REQUIRED, or VERIFY_USER. The only valid values are NO_CLIENT_AUTH, REQUIRED, and VERIFY_USER.

System Action: The line containing the error is ignored. FTP.DATA file processing continues with the next line of 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 statement. See the *z/OS Communications Server: IP Configuration Guide* for information about configuring the FTP server to use the TLS security mechanism.

Source Data Set: EZAFTPEP

Procedure Name: read_ftpdata

EZYFT99 **Domain name unknown. Getaddrinfo() rc** *getaddrinfo_rc*.

Explanation: The FTP server was not able to determine its domain name because the `getaddrinfo()` resolver call failed.

getaddrinfo_rc is the return code from `getaddrinfo()`. See the *z/OS XL C/C++ Run-Time Library Reference* for information about `getaddrinfo()` return codes.

System Action: FTP continues processing. The host domain name will not be known to the server.

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

System Programmer Response: Ensure that the host name is accessible through a name server, or defined in a local hosts or ipnodes data set. Refer to the *z/OS Communications Server: IP Configuration Guide* for information about domain name systems, local hosts data sets, and ipnodes data sets.

Source Data Set: `ezaftpsd`

Procedure Name: `handle_client_socket()`

Chapter 9. EZYPxxxx messages

EZYPR001 There were no records to satisfy the request

Explanation: No records matching the selection criteria were found.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQ

EZYPR002 Invalid Trace Option, please enter a 1 or 2

Explanation: The value specified in the Trace Option field is not valid.

System Action: The system continues processing.

User or Operator Response: Specify a valid value in the Trace Option field and reissue the request. A valid trace option is a "1" or "2".

System Programmer Response: None.

Source Data Set: EZAPPDGP

EZYPR003 Invalid Confirm option, please enter a "/" or clear field

Explanation: The value specified in the Confirm option field is not valid.

System Action: The system continues processing.

User or Operator Response: Specify a valid value in the Confirm option field and reissue the request. A valid confirm option is a "/" or blank.

System Programmer Response: None.

Source Data Set: EZAPPDGP

EZYPR004 Dataset name must be in valid dataset format

Explanation: The data entered for the data set name contains an invalid character.

System Action: The system continues processing.

User or Operator Response: Enter valid data for the data set name field. The data set name must follow the standard TSO data set naming rules. See *z/OS TSO/E User's Guide* for more information.

System Programmer Response: None.

Source Data Set: various

EZYPR005 Delete request for the record was canceled

Explanation: A delete request was issued for a selected record, but it was canceled by the user. As a result, the record is not deleted.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: various

EZYPR006 Queue record requested not found

Explanation: The requested queue file record cannot be processed because it does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPQ

EZYPR007 Table has mismatched record(s); record refresh was issued.

Explanation: For one or more queue file records, a requested operation was not performed because of a mismatch between the data shown on the NPF Queue List panel and the data currently in the record.

System Action: For each mismatched record, the requested operation is not performed; the data on the NPF Queue List panel is updated to match the current record data, and a '?' is displayed in the Action column.

User or Operator Response: For each mismatched record, check the updated data on the Queue List panel to determine if the requested operation is still needed. If so, repeat the request.

System Programmer Response: None.

Source Data Set: EZAPPQ.

EZYPR008 Incorrect call to EZAPPQ

Explanation: An invalid call to module EZAPPQ from panel EZAPPMP was issued. Expected parameters in the call may have been missing.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: EZAPPQ

EZYPR009 Cannot create queue table

Explanation: An error occurred during the creation of a queue table. The queue table is used for displaying the list of queue records.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: EZAPPQ

EZYPR010 Queue file open failed

Explanation: Either the queue file cannot be opened, access to the queue file is denied, or the queue file does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Verify the existence of the queue file. If it exists, then determine if the queue file is read-protected. If it is, consult your Resource Access Control Facility (RACF) administrator. If the queue file is not read-protected and the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPQ

EZYPR011 Queue file read failed

Explanation: An error occurred during the attempt to read a queue file record from the queue file.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: EZAPPQ

EZYPR014 Queue file update failed

Explanation: A requested update to a queue file record could not be completed because of an I/O error.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Determine the cause of the I/O error and take appropriate action.

Source Data Set: EZAPPQ

EZYPR015 Invalid return code from ISPF

Explanation: The ISPF dialog management services detected an error during panel processing.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: various

EZYPR016 Invalid command entered

Explanation: The data entered in the command line area is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Correct the command and reissue the request.

Source Data Set: various

EZYPR017 Trace file cannot be opened

Explanation: Either the trace file cannot be opened, the trace file cannot be dynamically allocated, access to the trace file is denied, or the trace file does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Verify the type of data set prefix used. If the prefix is a TSO user ID, the trace file should be dynamically allocated during the open attempt. If the dynamic allocation failed, contact the IBM Software Support Center. If the prefix is other than a TSO user ID, verify the existence of the trace file. If it exists, then determine if the trace file is write-protected. If it is, consult your Resource Access Control Facility (RACF) administrator. If the trace file is not write-protected and the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPDG

EZYPR018 You do not have access authorization for this dataset

Explanation: The data set is security protected for read or update by the Resource Access Control Facility (RACF). All access attempts to this data set will be denied.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact your Resource Access Control Facility (RACF) administrator if you want access permission to this data set.

Source Data Set: various

EZYPR019 You do not have update authorization for this dataset

Explanation: The data set is security protected for update by the Resource Access Control Facility (RACF). The data set is available for read-only. You cannot edit a record in the data set, but you can browse a record in the data set.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact your Resource Access Control Facility (RACF) administrator if you want update permission to this data set.

Source Data Set: various

EZYPR020 Last Send Time is not valid

Explanation: The value specified in the Last Send Time field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Last Send Time and reissue the request.

Source Data Set: EZAPPQRP

EZYPR021 Last Send Date is not valid

Explanation: The value specified in the Last Send Date field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Last Send Date and reissue the request.

Source Data Set: EZAPPQRP

EZYPR022 Printer Name must be specified

Explanation: The value specified in the Printer Name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Printer Name and reissue the request.

Source Data Set: various

EZYPR023 Host Name or IP address must be specified

Explanation: The value specified in the Host Name or IP address field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Host Name or IP address and reissue the request.

Source Data Set: various

EZYPR024 Command is not valid for the current record status

Explanation: An operation requested for a queue record was not performed because the request is not allowed for records with this data set status value.

System Action: The system continues processing.

User or Operator Response: Use a command that is valid for the current status of the queue record you are working with.

System Programmer Response: None.

Source Data Set: EZAPPQ.

EZYPR025 Record is currently in use; browse or retry later

Explanation: A requested operation cannot be performed at this time for a specified queue record file because the record is being updated by either the NPF Queue Manager or another panel operator.

System Action: The system continues processing.

User or Operator Response: Browse the queue file record if read-only access is sufficient; otherwise, retry the original operation later.

System Programmer Response: None.

Source Data Set: EZAPPQ.

EZYPR026 Queue record was changed; please re-enter your changes.

Explanation: An edit operation on a queue file record took too long (over 12 minutes) allowing the NPF Queue Manager or another panel operator to access and modify that record. All updates from the attempted edit operation have been lost and must be re-entered.

System Action: The system continues processing.

User or Operator Response: Repeat the edit operation, being careful to complete the operation within 12 minutes which is the maximum time for which you are guaranteed exclusive access to the record.

System Programmer Response: None.

Source Data Set: EZAPPQ.

EZYPR027 Queue record with status X can only be browsed

Explanation: An edit request has been rejected for a queue record whose data set status = 'X'.

System Action: The system continues processing.

User or Operator Response: Do not try to edit queue records whose data set status = 'X'. Browse is the only valid operation for this type of record.

System Programmer Response: None.

Source Data Set: EZAPPQ.

EZYPR028 Queue record with status X cannot be deleted

Explanation: A delete request has been rejected for a queue record whose data set status = 'X'.

System Action: The system continues processing.

User or Operator Response: Do not try to delete queue records whose data set status = 'X'. Browse is the only valid operation on this type of record.

System Programmer Response: None.

EZYPR030 • EZYPR034

Source Data Set: EZAPPQ

EZYPR030 Dataset type is in use by another dialog on this userid

Explanation: An attempt to process an NPF routing, options or queue file was rejected because another dialog under the same userid was already processing a file of the same type.

System Action: The system continues processing.

User or Operator Response: When doing NPF processing from multiple dialogs under a single userid, have each dialog process a different type of file. For example, it is acceptable to process a routing file from one dialog and an options file from another dialog.

System Programmer Response: None.

Source Data Set: EZAPPQ or EZAPPPC

EZYPR031 Options file record already exists.

Explanation: The specified new options record cannot be added to the options file because that record already exists.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a new options name for the key field other than the one that already exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR032 Options file record is not added.

Explanation: The options file record was not added, or you do not have write access to the options file.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Check the options file for correct information and reissue the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR033 Options file record is added.

Explanation: The new record was added to the options file as requested.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR034 Options file record does not exist.

Explanation: The system cannot delete the options file record you requested because it does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Select an existing options file record that you want deleted and reissue the request.

Source Data Set: EZAPPPC

EZYPR035 Options file record is not deleted.

Explanation: The options file record cannot be deleted.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Check the options file for correct information and reissue the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR036 Options file record is deleted.

Explanation: The options file record was deleted as requested.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR037 Type of option selected is not valid.

Explanation: The option selected is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Select a valid option and enter the request.

Source Data Set: various

EZYPR038 Missing broadcast rec(s) detected; Issue "CAncel" or complete

Explanation: More than one destination was specified for the normal route, and data has not been entered for the specific routing broadcasts.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Enter data for the specific routing broadcasts on the routing broadcast record panel. If a normal route and no specific routing broadcasts are wanted, then change the value in the number of destinations field to "1". If the data for normal and specific routing records are to be discarded, issue the cancel command on the command line.

Source Data Set: EZAPPPC

EZYPR041 Old options file record does not exist.

Explanation: The copy request cannot be completed because the old options file record specified does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify an old options file record that exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR042 New options file record is not copied.

Explanation: The new options file record was not copied.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Check the options file for correct information and reissue the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR043 New options file record is copied.

Explanation: The new options file record was copied as requested.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR044 Options file record does not exist.

Explanation: The requested options file record cannot be updated because it does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify an options file record that exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR045 Options file record is not updated.

Explanation: The options file record was not updated.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Check the options file for correct information and reissue the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR046 Options file record is updated.

Explanation: The options file record was updated as requested.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR048 New options file record already exists.

Explanation: The options file record cannot be copied because a new options file record already exists.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify an options file record other than the one that already exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR049 Select an option.

Explanation: An option must be selected from this panel.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Select an option from the panel and issue the request.

Source Data Set: various

EZYPR051 Enter options name for the key-field.

Explanation: The options name for the key-field information is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Enter a value in the options name for the key-field and reissue the request.

Source Data Set: EZAPPN1

EZYPR052 Input record exit is not valid.

Explanation: The value specified in the input record exit field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid input record exit value and reissue the request.

Source Data Set: EZAPPN7

EZYPR053 Old options name is not valid.

Explanation: The value specified in the old options name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid value in the old options name field and reissue the request.

Source Data Set: EZAPPN4

EZYPR054 New options name is not valid.

Explanation: The value specified in the new options name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid value in the new options name field and reissue the request.

Source Data Set: EZAPPN4

EZYPR055 Tracing is enabled

Explanation: The request to enable NPF ISPF tracing is acknowledged, and the specified trace data set is opened to receive trace information.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPDG

EZYPR056 Tracing is disabled

Explanation: The request to disable NPF ISPF tracing is acknowledged, and the specified trace data set is closed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPDG

EZYPR057 Old options name must be specified.

Explanation: The old options name for the key-field is missing.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Enter a value for the old options name key-field and reissue the request.

Source Data Set: EZAPPN4

EZYPR058 New options name must be specified.

Explanation: The new options name for the key-field is missing.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a value for the new options key-field and reissue the request.

Source Data Set: EZAPPN4

EZYPR060 Press ENTER to save data before proceeding to next record

Explanation: During add processing for a multiple-destination routing, the DOWN PF key was pressed to proceed to the next destination record before the ENTER key was pressed to save the data for the current destination record.

System Action: The DOWN PF key is ignored and the panel continues to display the current destination record.

User or Operator Response: Press the ENTER key to save the data for the current destination record before attempting to process the next record.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR061 Normal route record already exists.

Explanation: The specified new route cannot be added to the routing file because that route already exists.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a normal route other than the one that already exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR062 Normal route record is not added.

Explanation: The normal route was not added.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR063 Normal route record is added.

Explanation: A normal route was added to the routing file.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR064 Normal route record does not exist.

Explanation: The route specified in the delete request does not exist, so the request cannot be completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a route that exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR065 Normal route record is not deleted.

Explanation: The normal route was not deleted, or you do not have write access to the routing file.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR066 Normal route record is deleted.

Explanation: The normal route is deleted from the routing file as requested.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR068 Routing file cannot be opened.

Explanation: Either the routing file cannot be opened, access to the routing file is denied, or the routing file does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Verify the existence of the routing file. If it exists, then determine if the routing file is read-protected. If it is, consult your Resource Access Control Facility (RACF) administrator. If the routing file is not read-protected and the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR069 Routing file cannot be closed.

Explanation: The system detected an error. The routing file cannot be closed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR071 Old normal route record does not exist.

Explanation: The normal route record cannot be copied because the old normal route does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify major and minor names for an old normal route record that exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR072 Normal route record is not copied

Explanation: The normal route record was not copied.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Reissue the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR073 Normal route record is copied

Explanation: The request to copy the normal route record is complete.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR074 Specific broadcast record does not exist.

Explanation: A specific broadcast record cannot be updated because it does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Because the specific broadcast record does not exist, all of the routing records associated with this missing broadcast record are not valid. Issue the delete request to delete the obsolete records and then issue the add request to create the new routing records.

Source Data Set: EZAPPPC

EZYPR075 Normal route record is not updated

Explanation: The normal route record was not updated.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Reissue the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR076 Normal route record is updated.

Explanation: The normal route record was updated as requested.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR078 New normal route record already exists.

Explanation: The normal route record cannot be copied because a new normal route record already exists.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify major and minor names for a routing file record other than the one that already exists and reissue the request.

Source Data Set: EZAPPPC

EZYPR079 Old specific broadcast record does not exist.

Explanation: The old specific broadcast record associated with the normal routing record in the request does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Since the old specific broadcast record does not exist, all of the routing records associated with this missing broadcast record are not valid. Issue the delete request to delete the obsolete records. Either specify major and minor names for a different routing record and reissue the request again or issue an add request to create new routing records which can be used for the copy request again.

Source Data Set: EZAPPPC

EZYPR081 **Enter old major name for the key-field.**

Explanation: The request is missing an old major name.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify an old major name and reissue the request. A valid major name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR082 **Enter new major name for the key-field.**

Explanation: The request is missing a new major name.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a new major name and reissue the request. A valid major name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR083 **Enter old minor name for the key-field.**

Explanation: The request is missing an old minor name for the key-field.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify an old minor name and reissue the request. A valid minor name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR084 **Enter new minor name for the key-field.**

Explanation: The request is missing a new minor name for the key-field.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a new minor name and reissue the request. A valid minor name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR086 **Enter major name for the key-field**

Explanation: The major name is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a major name and reissue the request. A valid major name is an 8-byte alphanumeric name.

Source Data Set: various

EZYPR087 Enter minor name for the key-field.

Explanation: The minor name is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a minor name and reissue the request. A valid minor name is an 8-byte alphanumeric name.

Source Data Set: various

EZYPR089 Number of destination(s) must be specified.

Explanation: The destination number is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a destination value and reissue the request. A valid value is in the range 1 to 65535.

Source Data Set: EZAPPN15

EZYPR091 Retain time (successful) must be specified

Explanation: The retain time (successful) is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify the retain time (successful) and reissue the request. Retain time must be in the format *dddhhmm* with values as follows:

ddd 000 to 366

hh 00 to 23

mm 00 to 59

Source Data Set: EZAPPN15

EZYPR092 Retain time (unsuccessful) must be specified.

Explanation: The retain time (unsuccessful) is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify the retain time (unsuccessful) and reissue the request. Retain time must be in the format *dddhhmm* with values as follows:

ddd 000 to 366

hh 00 to 23

mm 00 to 59

Source Data Set: EZAPPN15

EZYPR093 Retry interval time must be specified.

Explanation: The retry interval time is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

EZYPR094 • EZYPR099

System Programmer Response: Specify the retry interval time and reissue the request. Retry interval time must be in the format *dddhlmm* with values as follows:

ddd 000 to 366

hh 00 to 23

mm 00 to 59

Source Data Set: EZAPPN15

EZYPR094 **Retry limit must be specified.**

Explanation: The retry limit is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a retry limit and reissue the request. A valid retry limit is in the range 0 to 65535.

Source Data Set: EZAPPN15

EZYPR095 **Options name must be specified.**

Explanation: The options name is missing from the request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify an options name and reissue the request. A valid options name is a 16-character alphanumeric name.

Source Data Set: EZAPPN15

EZYPR097 **Specific broadcast record is not updated.**

Explanation: The system detected an error. The specific broadcast record update request was not completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR098 **Specific broadcast record is updated.**

Explanation: The specific broadcast record update request completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR099 **Specific broadcast record already exists.**

Explanation: The specific broadcast record cannot be added to the routing file because that route already exists.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Because the specific broadcast record already exists, the created normal routing

record is not valid. Issue the delete request to delete the obsolete records. Issue an add request to create the new routing records again.

Source Data Set: EZAPPPC

EZYPR101 Specific broadcast record is added.

Explanation: A specific broadcast route was added to the routing file.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR102 Specific broadcast is not added.

Explanation: The system detected an error. The specific broadcast was not added.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Restart the Network Print Facility and reissue the add request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR104 Number of destination(s) is not valid

Explanation: The number of destinations value must be larger than 0.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid value for the number of destinations. A valid value is from 1 to 65535.

Source Data Set: various

EZYPR105 Specific broadcast record is not deleted.

Explanation: The system detected an error. The specific broadcast record delete request was not completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR106 Normal route & specific broadcast record(s) are deleted.

Explanation: The normal route and specific broadcast records delete request completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR107 Specific broadcast record is not copied.

Explanation: The system detected an error. The specific broadcast record copy request was not completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR108 Normal route & specific broadcast record(s) are copied.

Explanation: The normal route and specific broadcast records copy request completed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR111 LU class is not valid.

Explanation: The LU class value entered is not valid. Valid values are 1 to 64.

System Action: The system continues processing.

User or Operator Response: Specify a valid LU class value and reissue the request.

System Programmer Response: None.

Source Data Set: various

EZYPR112 LU class contains invalid character.

Explanation: A character in the LU class field is not valid. Valid values are 1 to 64.

System Action: The system continues processing.

User or Operator Response: Specify a valid LU class value and reissue the request.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR113 Incorrect call to EZAPPPC

Explanation: The call to module EZAPPPC from panel EZAPPMP was not valid. Expected parameters in the call may have been missing.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR116 Normal route & specific broadcast record(s) are not added

Explanation: The normal and/or specific broadcast records were added during the add request, but a cancel command was issued to discard the added records.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAPPPC

EZYPR117 Normal route is copied but some data is lost.

Explanation: The normal route copy request completed, but not all of the data was copied.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Submit the copy request again. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR131 Old major name is not valid.

Explanation: The value specified in the old major name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid old major name and reissue the request. A valid old major name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR132 Old minor name is not valid.

Explanation: The value specified in the old minor name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid old minor name and reissue the request. A valid old minor name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR133 New major name is not valid.

Explanation: The value specified in the new major name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid new major name and reissue the request. A valid new major name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR134 New minor name is not valid.

Explanation: The value specified in the new minor name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid new minor name and reissue the request. A valid new minor name is an 8-byte alphanumeric name.

Source Data Set: EZAPPN12

EZYPR135 Major name is not valid.

Explanation: The value specified in the major name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid major name and reissue the request. A valid major name is an 8-byte alphanumeric name.

Source Data Set: various

EZYPR136 Minor name is not valid.

Explanation: The value specified in the minor name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid minor name and reissue the request. A valid minor name is an 8-byte alphanumeric name.

Source Data Set: various

EZYPR141 Options name is not valid.

Explanation: The value specified in the Options name field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid value in the Options name field and reissue the request. A valid options name is a 16-character alphanumeric name.

Source Data Set: various

EZYPR142 Routing exit is not valid.

Explanation: The value specified in the Routing exit field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid value in the Routing exit field and reissue the request. A valid routing exit name is an 8-character alphanumeric name.

Source Data Set: EZAPPN15

EZYPR143 Normal route & specific broadcast record(s) not deleted

Explanation: The system detected an error. The normal and specific broadcast records were not deleted as a result of the delete request.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Retry the request. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR144 Unable to copy old specific broadcast record.

Explanation: The system detected an error. A specific broadcast record associated with the normal routing record could not be copied.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Because the old specific broadcast record is not valid, all of the copied routing records with this broadcast record are deleted. Issue the delete request to delete the obsolete records for the old routing records. Either specify major and minor names for a different routing record and reissue the request again, or issue an add request to create new routing records which can be used for the copy request again.

Source Data Set: EZAPPPC

EZYPR147 Unable to copy old normal route record.

Explanation: The system detected an error. A copy cannot be done for the old normal route record specified.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Try the request again. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR148 Normal route record is added but some data is lost.

Explanation: The normal route record add request completed, but not all of the data was added.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Submit the add request again. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR149 Specific broadcast record is added but some data is lost.

Explanation: The specific broadcast record add request completed, but not all of the data was added.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Submit the add request again. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR161 Options file cannot be opened.

Explanation: Either the options file cannot be opened, access to the options file is denied, or the options file does not exist.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Verify the existence of the options file. If it exists, then determine if the options file is read-protected. If it is, consult your Resource Access Control Facility (RACF) administrator. If the routing file is not read-protected and the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPPC

EZYPR162 Options file cannot be closed.

Explanation: The system detected an error. The options file specified cannot be closed.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Try the request again. If the error still occurs, contact the IBM Software Support Center.

Source Data Set: EZAPPRT

EZYPR180 SLU/Dest is not valid

Explanation: The value specified in the SLU/Dest field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid SLU/Dest value and reissue the request.

Source Data Set: EZAPPQSP

EZYPR181 PLU/JOBNAME is not valid

Explanation: The value specified in the PLU/JOBNAME field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid PLU/JOBNAME value and reissue the request.

Source Data Set: EZAPPQSP

EZYPR182 Creation Date is not valid

Explanation: The value specified in the Creation Date field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Creation Date and reissue the request.

Source Data Set: EZAPPQSP

EZYPR183 Creation Time is not valid

Explanation: The value specified in the Creation Time field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Creation Time and reissue the request.

Source Data Set: EZAPPQSP

EZYPR184 Dataset Status is not valid

Explanation: The value specified in the Dataset Status field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Dataset Status and reissue the request.

Source Data Set: EZAPPQSP

EZYPR187 Invalid SEQ value - enter KEY or TIME

Explanation: The SEQ field on the NPF Queue Selection panel contains an invalid value.

System Action: The system continues processing.

User or Operator Response: Set the SEQ field to either KEY or TIME to indicate the order in which the selected queue records should be displayed on the Queue List panel.

System Programmer Response: None.

Source Data Set: EZAPPQ

EZYPR190 Next Send Time is not valid

Explanation: The value specified in the Next Send Time field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Next Send Time value and reissue the request.

Source Data Set: EZAPPQSP

EZYPR191 Next Send Date is not valid

Explanation: The value specified in the Next Send Date field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Next Send Date value and reissue the request.

Source Data Set: EZAPPQSP

EZYPR192 Retry Interval is not valid

Explanation: The value specified in the Retry Interval field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Retry Interval value and reissue the request.

Source Data Set: various

EZYPR193 Retry Attempts are not valid

Explanation: The value specified in the Retry Attempts field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Retry Attempts value and reissue the request.

Source Data Set: EZAPPQRP

EZYPR194 Retry Limit is not valid

Explanation: The value specified in the Retry Limit field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Retry Limit value and reissue the request.

Source Data Set: various

EZYPR195 Retain Time Successful is not valid

Explanation: The value specified in the Retain Time Successful field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Retain Time Successful value and reissue the request.

Source Data Set: various

EZYPR196 Retain Time Unsuccessful is not valid

Explanation: The value specified in the Retain Time Unsuccessful field is not valid.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: Specify a valid Retain Time Unsuccessful value and reissue the request.

Source Data Set: various

EZYPR213 Default page format name is not valid

Explanation: The value specified in the Def Page Format field is not valid.

System Action: The system continues processing.

User or Operator Response: Either leave the field blank or enter the name of a valid Default Page Format table entry. See the *z/OS Communications Server: IP Network Print Facility* for more information.

System Programmer Response: None.

Source Data Set: EZAPPN15

EZYPR214 End of file name is not valid.

Explanation: The value entered for Eofile Name field is not valid.

System Action: The system continues processing.

User or Operator Response: Either leave the field blank or enter the name of a valid End-of-File Rules table entry. See the *z/OS Communications Server: IP Network Print Facility* for more information.

System Programmer Response: None.

Source Data Set: EZAPPN15

Chapter 10. EZYRxxxx messages

EZYRC01I Calling function rexec with the following:

Explanation: This is the first of a two part message. It is followed by EZYRC02I which lists the host, user, cmd, and port that are being passed to the rexec function that is provided by z/OS UNIX System Services and that does the work of passing the rexec information to the server.

System Action: OREXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC02I Host: AAA user BBB, cmd CCC, port DDD

Explanation: This is the information that is being passed to the rexec function. AAA is the host that the string is being passed to, BBB is the logon ID of the user on the remote system, CCC is the command that is being passed to the remote system and DDD is the port number.

System Action: OREXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC03E The call to rexec procedure failed.

Explanation: The call to the rexec procedure that will do the actual passing of the information to the remote host failed.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent. If inetd is down on the remote host, this message will appear.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC04I Usage: orexec -d -l user -p pwd

Explanation: This is the first of a group of messages that is issued when a question mark has been entered, or an invalid number of arguments have been entered.

System Action: OREXEC ends.

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

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC05I **options: -**

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC06I **-? display this message**

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC07I **-d turn on debug tracing**

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC08I **-l *usr* specifies remote login id**

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC09I **-p *pwd* specifies remote password**

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC10I -s port specifies server port

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC11I -V display APAR level

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC12I Example: orexec -d -l guest -p guest hostname ls -l

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: By examining the command typed or using the -d option the problem should be apparent.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC13I XXX YYY

Explanation: XXX is the program called (MVS OREXEC).

YYY is the APAR number, or if there have been no APARs applied, it is base.

System Action: OREXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC14I -s port fhost command

Explanation: This the second of a group of messages that is issued when a question mark has been entered, or an invalid number of arguments have been entered.

System Action: OREXEC ends.

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

EZYRC15I • EZYRC18I

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC15I -C Uppercase messages

Explanation: This is part of a group of messages that is issued when a question mark has been entered or an invalid number of arguments have been entered.

System Action: OREXEC ends.

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

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC16E *function failed. description* rsn = *errnojr*

Explanation: An unexpected value was returned from *function*. Processing ends.

function is the name of the C/C++ run-time function. See the *z/OS XL C/C++ Run-Time Library Reference* for information about C/C++ run-time functions.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OREXEC ends.

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

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC17I Select time limit expired.

Explanation: The time limit on the select expired before a response was received from the server. Processing ends.

System Action: OREXEC ends.

User or Operator Response: Attempt the OREXEC command again.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC18I Foreign host name is missing.

Explanation: The foreign host name was not specified on the OREXEC command. It is required. Processing ends.

System Action: OREXEC ends.

User or Operator Response: Attempt the OREXEC command again with the foreign host name.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC19I **Data socket = *dsock*, Control socket = *csock*.**

Explanation: This is a debug message indicating the socket descriptors being used by the z/OS UNIX REXEC client. It is only issued when the -d option is specified on the OREXEC command.

System Action: OREXEC continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC23E **Command is missing**

Explanation: A command was not specified on the OREXEC command.

System Action: OREXEC ends.

User or Operator Response: Reissue the OREXEC command and specify a command.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: main()

EZYRC24I **-e *wait* select time limit**

Explanation: See message EZYRC04I.

System Action: OREXEC ends.

User or Operator Response: Examine the command typed or use the -d option to diagnose the problem.

System Programmer Response: None.

Source Data Set: rexec1.c

Procedure Name: usage()

EZYRC31I **Calling function rcmd_af with the following:**

Explanation: This message is followed by EZYRC02I, which lists the host, user, cmd, and port that are passed to the local rcmd_af function, which passes the rsh information to the server.

System Action: ORSH continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ussrsh.c

Procedure Name: main()

EZYRC33E **The call to rcmd_af procedure failed: *description* rsn = *errnojr***

Explanation: The call to the rcmd_af procedure that will pass the information to the remote host failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System Action: ORSH ends.

User or Operator Response: Examine the command typed and reissue the ORSH command. If the problem persists,

EZYRD01W • EZYRD04E

contact the system programmer. The most common description is ECONNREFUSED. If this occurs, start the remote shell server on the remote host.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: ussrsh.c

Procedure Name: main()

EZYRD01W Invalid option in /etc/inet.conf

Explanation: An option that is not valid has been used in the /etc/inetd.conf file. The allowable options are: dIV

Note: If an invalid option has been specified, none of the valid options will be in effect.

System Action: REXECD continues.

User or Operator Response: None.

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

Source Data Set: rexecd.c

Procedure Name: main

EZYRD02E getpeername failure

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

System Action: REXECD ends.

User or Operator Response: Retry the command later.

System Programmer Response: See message EZYRD33E on the host where rexecd is running for more information including the failure description (errno) and reason code (errnojr).

Source Data Set: rexecd.c

Procedure Name: main

EZYRD03I Remote address = XX.XX.XX.XX

Explanation: XX.XX.XX.XX is the IPv4 or IPv6 address that is the internet address that the login ID connected from. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: main

EZYRD04E Binary one not received.

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

System Action: REXECD ends.

User or Operator Response: Retry the command.

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD05I clisecport = n

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

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD06E Unable to create secondary port.

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

System Action: REXECD ends.

User or Operator Response: Retry the command.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD07E Cannot make second port

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

System Action: REXECD ends.

User or Operator Response: Retry the command.

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

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

Source Data Set: rexecd.c

EZYRD08I • EZYRD11E

Procedure Name: doit()

EZYRD08I User is: XXX

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

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD09I Command is: XXX

Explanation: XXX is the command that has been passed from the remote host. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD10E XXX: unknown login. cmd = YYY

Explanation: XXX is the logon ID that was passed in from the remote host.

YYY is the command that was passed in.

System Action: REXECD ends.

User or Operator Response: Correct the login ID and retry.

System Programmer Response: This message is written to the file that is specified by syslog.conf for information and authorization messages. It is only written out if either information and/or authorization is in the syslogd.conf file. This indicates that the login ID was not in the password structure.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD11E Login incorrect.

Explanation: The login was incorrect.

System Action: REXECD ends.

User or Operator Response: Retry the command.

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

Source Data Set: rexecd.c

Procedure Name: doit()

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

Explanation: This is a debug message. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

XXX is the login ID that has been passed in from the client.

YYY is the login ID as gotten from the password structure.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD13I **dir is XXX**

Explanation: XXX is the home directory that is gotten from the password structure. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

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

Explanation: X is the uid that is obtained from the password structure and Y is the gid that is obtained from the password structure. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

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

Explanation: XX is the login ID passed in from the client and YY is the command that is being attempted. The password that has been passed in is not correct. This message is written to the file that is specified by syslog.conf for error messages. It is only written out if error is set up to write out in syslogd.

System Action: REXECD ends.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD17E XX no home directory. cmd = YY

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

System Action: REXECD ends.

User or Operator Response: Contact the system programmer

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD18E No remote directory.

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

System Action: REXECD ends.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD19E Cannot make pipe1.

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

System Action: REXECD ends.

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

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD20E Cannot make pipe2.

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

System Action: REXECD ends.

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

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD21E Cannot make pipe3.

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

System Action: REXECD ends.

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

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD22E Cannot fork; try again.

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

System Action: REXECD ends.

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

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD23E error on read socket 0

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

System Action: REXECD ends.

User or Operator Response: Retry the failing command.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD24I XX cmd = YY

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

System Action: REXECD continues.

User or Operator Response: None.

EZYRD25E • EZYRD28E

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD25E setgid failed

Explanation: A set group ID subroutine failed.

System Action: REXECD ends.

User or Operator Response: Contact the system programmer.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD26E initgroups failed

Explanation: An initgroups subroutine failed.

System Action: REXECD ends.

User or Operator Response: Contact the system programmer.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD27E setuid failed

Explanation: A setuid failed.

System Action: REXECD ends.

User or Operator Response: Contact the systems programmer.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD28E Error from execl()

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

System Action: REXECD ends.

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

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD29E String too long

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

System Action: REXECD ends.

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

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

Source Data Set: rexecd.c

Procedure Name: getstr()

EZYRD30I usage: rexecd -dLV

Explanation: This message follows EZYRD01. It displays the valid options that are permitted in /etc/inetd.conf for the rexecd server.

System Action: REXECD continues.

User or Operator Response: None.

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

Source Data Set: rexecd.c

Procedure Name: usage()

EZYRD31I XXX YYY

Explanation: XXX is the program called (MVS REXECD).

YYY is the APAR number, or if there have been no APARs applied, it is base.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: main()

EZYRD32E Cannot make second port

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

System Action: REXECD ends.

User or Operator Response: Retry the command.

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

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

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD33E *function failed. description* rsn = *errnojr*

Explanation: An unexpected value was returned from *function*.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojs) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: REXECD ends.

User or Operator Response: Reissue the REXEC command, if the problem persist contact the system programmer.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: rexecd.c

Procedure Name: main()

EZYRD34I Stderr socket is *socket*

Explanation: Shows the socket number used for the standard error connection. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD35I **Int(socket):** 20-bytes hex data 10 char ascii

Explanation: Shows the data being sent from the client to the server after the command has been read. *socket* is the socket from which the data was read. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD36I **Ont(socket):** 20-bytes hex data 10 char ascii

Explanation: Shows the data being sent from the server to the client after the command has been read. *socket* is the socket to which the data was written. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD37I **Signal signal received on socket socket**

Explanation: Shows the value of a signal received from the client. This signal will be passed to the child process executing the command. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD38I **EOF received on socket socket**

Explanation: Indicates a EOF was received on the socket. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD39I **Child stderr socket is socket**

Explanation: Identifies the socket assigned in the parent process to the child standard error. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rexecd and debug is set up to write out in syslogd.

EZYRD42I • EZYRP37E

System Action: REXECD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rexecd.c

Procedure Name: doit()

EZYRD42I getsockname failure

Explanation: Getsockname could not return the local address structure that is connected to the socket.

System Action: REXECD ends.

User or Operator Response: Retry the command later.

System Programmer Response: See message EZYRD33E on the host where rexecd is running for more information including the failure description (errno) and reason code (errnojr.)

Source Data Set: rexecd.c

Procedure Name: main

EZYRP35I orpcinfo -d prognum versnum

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

System Action: ORPCINFO exits.

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

System Programmer Response: None.

Source Data Set: ORPCINFO

Procedure Name: Usage()

EZYRP36E Sorry. You are not root

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

System Action: The registration is not deleted.

User or Operator Response: None.

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

Source Data Set: orpcinfo

Procedure Name: stderr

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

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

System Action: The registration is not deleted.

User or Operator Response: None.

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

Source Data Set: orpcinfo

Procedure Name: stderr

EZYRP52E oportmap CALLIT: cannot fork

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

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

User or Operator Response: None.

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

Source Data Set: oportmap

Procedure Name: stderr

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

Explanation:

System Action: The server continues.

User or Operator Response:

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP67E enablecache: could not allocate cache

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

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

User or Operator Response: None.

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP68E enablecache: could not allocate cache data

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

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

User or Operator Response: None.

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP69E enablecache: could not allocate cache fifo

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

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

User or Operator Response: None.

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP70E cache_set: victim not found

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

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

User or Operator Response: None.

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP71E cache_set: victim alloc failed

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

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

User or Operator Response: None.

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRP72E cache_set: could not allocate new rpc_buffer

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

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

User or Operator Response: None.

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

Source Data Set: svc_upd.o

Procedure Name: stderr

EZYRS01I XXX YYY

| **Explanation:** XXX is the program called (MVS RSHD).

YYY is the APAR number, or if there have been no APARs applied, it is base.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS02W Invalid option in /etc/inet.conf

Explanation: An invalid option has been used in the /etc/inetd.conf file. The allowable options are: dlv NOTE: If an invalid option has been specified None. of the valid options will be in effect.

System Action: RSHD continues.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: main

EZYRS03E Getpeername failure.

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

System Action: RSHD ends.

User or Operator Response: Retry the command later.

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

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

Source Data Set: rshd.c

Procedure Name: main

EZYRS04W Setsockopt (SO_KEEPALIVE) failed

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

System Action: RSHD continues.

User or Operator Response: Retry the command.

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

Source Data Set: rshd.c

Procedure Name: main

EZYRS05W Setsockopt (SO_LINGER) failed

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

System Action: RSHD continues.

User or Operator Response: Retry the command.

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

Source Data Set: rshd.c

Procedure Name: main

EZYRS06E Malformed from address.

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

System Action: RSHD ends.

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

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit

EZYRS07W Connection received using IP options (ignored)

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

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit

EZYRS08E Setsockopt IP_OPTIONS NULL.

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit

EZYRS09E Connection from XX on illegal port YY

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

System Action: RSHD ends.

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

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit

EZYRS10E Connection on illegal port.

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

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

System Action: RSHD ends.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit

EZYRS11E Binary one not received.

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

System Action: RSHD ends.

User or Operator Response: Retry the command.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS12I clisecport = n

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

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS13E Cannot get stderr port

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

System Action: RSHD ends.

User or Operator Response: Try again later.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS16E Second port not reserved

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS18E Cannot make second port

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

System Action: RSHD ends.

User or Operator Response: Retry the command.

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

- The Socket parameter is not valid.
- The Socket parameter refers to a file, not a socket.
- The specified address is not available from the local machine.
- The addresses in the specified address family cannot be used with this socket.
- The socket is already connected.
- The establishment of a connection timed out before a connection was made.
- The attempt to connect was rejected.
- No route to the network or host is present.
- The specified address is already in use.

EZYRS19E • EZYRS20I

- The Address parameter is not in a writable part of the user address space.
- The socket is marked as nonblocking. The connection cannot be immediately completed. The application program can select the socket for writing during the connection process.
- The specified path name contains a character with the high-order bit set.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS19E Cannot make second port

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

System Action: RSHD ends.

User or Operator Response: Retry the command.

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS20I Could not look up address for XXX

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

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

System Action: RSHD ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSHD.c

Procedure Name: doit()

EZYRS21I Remote user is: XXX

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

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSHD.c

Procedure Name: doit()

EZYRS22I Local user is: XXX

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

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RSHD.c

Procedure Name: doit()

EZYRS23I Command is: XXX

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

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

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

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS25E Unknown login.

Explanation: The login was incorrect.

System Action: RSHD ends.

User or Operator Response: Retry the command.

EZYRS26E • EZYRS30E

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

Source Data Set: rshd.c

Procedure Name: doit()

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

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS27E **No remote directory.**

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

System Action: RSHD ends.

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

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

Source Data Set: RSHD.c

Procedure Name: doit()

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

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS30E **Logins currently disabled.**

Explanation: The systems administrator has disabled logins.

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS31E Cannot make pipe.

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

System Action: RSHD ends.

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

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS32E Cannot make pipe1.

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

System Action: RSHD ends.

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

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS33E Cannot make pipe2.

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

System Action: RSHD ends.

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

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS34E Cannot fork; try again.

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

System Action: RSHD ends.

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

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS35E

Explanation: An error occurred while reading data from socket zero. Socket zero is the standard input socket.

System Action: RSHD ends.

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS36I WWW@XXX as YYY: cmd = zzz

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

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS37E setgid failed

Explanation: A set group ID subroutine failed.

System Action: RSHD ends.

User or Operator Response: Contact the system programmer.

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS38E initgroups failed

Explanation: An initgroups subroutine failed.

System Action: RSHD ends.

User or Operator Response: Contact the system programmer.

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS39E setuid failed

Explanation: A setuid failed.

System Action: RSHD ends.

User or Operator Response: Contact the systems programmer.

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS40E

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

System Action: RSHD ends.

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

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

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

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS41E String too long

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

System Action: RSHD ends.

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

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

Source Data Set: rshd.c

Procedure Name: getstr()

EZYRS42E **rshd -adlnLV**

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: getstr()

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

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

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

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

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

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

System Action: RSHD ends.

User or Operator Response: None.

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

Source Data Set: rshd.c

Procedure Name: doit()

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

Explanation: If the -L option has been specified and if the syslog.conf file has been set up to send information or authorization messages to a file then this message will appear in the file. The remote user WWW logging on as user ID YYY has executed command zzzq.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS47I WWW as YYY: cmd = zzz

Explanation: If the -L option has been specified and if the syslog.conf file has been set up to send information or authorization messages to a file then this message will appear in the file. The remote user WWW logging on as user ID YYY has executed command zzz.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS48I Trusted host activated

Explanation: The rsh client has issued a command that does not have a password in it. The trusted host installation exit has been invoked. This message is only written to the log file if syslogd.conf has been configured and LOG_INFO has been specified.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS49I Trusted host authentication failed.

Explanation: The rsh client has issued a command that does not have a password in it. The trusted host installation exit has been invoked. The installation exit returned something other than a zero (0) return code. This message will be followed by message EZYRS36I. These messages are only written to the log file if syslogd.conf has been configured and LOG_INFO has been specified.

System Action: RSHD ends.

User or Operator Response: The .rhosts file has incorrect information or there has been an attempt to circumvent the authentication.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS50E *function* failed. *description* rsn = errnojr

Explanation: An unexpected value was returned from *function*. Processing ends.

function is the name of the C/C++ run-time function. See the z/OS XL C/C++ Run-Time Library Reference for information about C/C++ run-time functions.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in

the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: RSHD ends.

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

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: rshd.c

Procedure Name: main()

EZYRS51I Stderr socket is *socket*

Explanation: Shows the socket number used for the standard error connection. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS52I Int(*socket*): 20-bytes hex data 10 char ascii

Explanation: Shows the data being sent from the client to the server after the command has been read. *socket* is the socket from which the data was read. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS53I Ont(*socket*): 20-bytes hex data 10 char ascii

Explanation: Shows the data being sent from the server to the client after the command has been read. *socket* is the socket to which the data was written. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS54I Signal *signal* received on socket *socket*

Explanation: Shows the value of a signal received from the client. This signal will be passed to the child process executing the command. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS55I EOF received on socket *socket*

Explanation: Indicates a EOF was received on the socket. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS56I Child stdin socket is *socket*

Explanation: Identifies the socket assigned in the parent process to the child standard input. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS57I Child stdout socket is *socket*

Explanation: Identifies the socket assigned in the parent process to the child standard output. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS58I Child stderr socket is *socket*

Explanation: Identifies the socket assigned in the parent process to the child standard error. This message is written to the file that is specified by syslog.conf for debug messages. It is only written out if both debug is turned on for rshd and debug is set up to write out in syslogd.

System Action: RSHD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: doit()

EZYRS59I Kerberos function *funcname* return code *rcode*

Explanation: This message is issued for debugging, error conditions, or both. It is written to syslog. It might be returned to the orshd client. If debugging is on for orshd, this message indicates that the Kerberos run-time function completed and gives the return code. If debugging is off for orshd, this message is issued if the Kerberos run-time function completes with a nonzero return code.

funcname is the Kerberos run-time function. Kerberos run-time functions are documented in the *z/OS Integrated Security Services Network Authentication Service Administration*.

rcode is the hexadecimal return code from the Kerberos run-time function. The return codes from the run-time functions are documented in the *z/OS Integrated Security Services Network Authentication Service Administration* under Kerberos Runtime Codes.

System Action: orshd ends.

User or Operator Response: If the message has a nonzero return code, the orshd client might not be authorized for kerberos use of orshd. Check with your system administrator.

System Programmer Response: Verify that the orsh client is authorized for Kerberos V5. Verify the Kerberos configuration.

Source Data Set: rshdkerb

Procedure Name: do_kerb5()

EZYRS60E Kerberos error - *errortype*

Explanation: This message is issued to syslog when the orshd server detects an error in the Kerberos communication with the client.

errortype can be one of the following:

V1 validation

In the Kerberos exchange between client and server, the V1 validation failed.

V2 validation

In the Kerberos exchange between client and server, the V2 validation failed.

AP REQ validation

In the Kerberos exchange between client and server, the AP REQ validation failed.

checksum failed

The orshd server start up configuration options specify that the Kerberos checksum must be valid (using -m option). In the Kerberos exchange between client and server, the checksum was not valid.

encryption required

The orshd server start up configuration options specify that kerberos and encryption will be used (using -e option). In the exchange between client and server, the client did not specify encryption.

forwarding is unsupported

Credentials forwarding was requested by the orsh Kerberos client. The orshd server does not support credentials forwarding.

System Action: orshd ends.

User or Operator Response: Contact the system or network administrator.

System Programmer Response: Turn on debugging for orshd to determine where in the Kerberos flow the error occurred. Verify that the orsh client is a valid Kerberos V5 client. There is probably a mismatch in the Kerberos support or options supported between the orsh client and server. Review the Kerberos configuration between the client and the server.

Source Data Set: rshdkerb

Procedure Name: do_kerb5()

EZYRS61E GSS-API function *funcname* error - *errmsg*

Explanation: This message is written to syslog by the orshd server. The Kerberos GSS-API run-time function failed. The text error message is provided.

funcname is the GSS-API run-time function that completed in error. GSS-API run-time functions are documented in the *z/OS Integrated Security Services Network Authentication Service Programming*.

errmsg is the text error message returned by the GSS-API run-time library. The messages from the run-time functions are documented in the *z/OS Integrated Security Services Network Authentication Service Administration*.

System Action: orshd ends.

User or Operator Response: The orshd client might not be authorized for GSS-API use of orshd. Contact your system administrator.

System Programmer Response: Turn on rshd debugging to determine where in the flow the error occurred. Verify that the client supports and is authorized for GSS. Verify the Kerberos GSS configuration between the client and server.

Source Data Set: rshdgss

Procedure Name: do_gssapi()

EZYRS62E GSS-API function CALLING error *call_error* ROUTINE error *routine_error*

Explanation: This orshd message accompanies EZYRS61E. When a GSS-API run-time function has a nonzero return code, the message is written to syslog. This message identifies the GSS-API CALLING error and the ROUTINE error as defined by the GSS-API, and might be used for debugging.

call_error is the calling error part of the major status value returned by the GSS-API run-time library function. The major status value is described in the *z/OS Integrated Security Services Network Authentication Service Programming*.

routine_error is the routine error part of the major status value returned by the GSS-API run-time library function. The major status value is described in the *z/OS Integrated Security Services Network Authentication Service Programming*.

System Action: orshd ends.

User or Operator Response: The orshd client might not be authorized for GSS-API use of orshd. Contact your system administrator.

System Programmer Response: Verify that the client supports and is authorized for GSS. Verify the Kerberos GSS configuration between the client and server.

Source Data Set: rshdgss

Procedure Name: do_gssapi()

EZYRS63I GSS-API function *funcname* complete

Explanation: This is an orshd informational message written to syslog if orshd has debugging turned on. It is used to determine which of the GSS-API run-time functions have completed successfully.

funcname is the GSS-API run-time function that completed. GSS-API run-time functions are documented in the *z/OS Integrated Security Services Network Authentication Service Programming*.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshdgss

Procedure Name: do_gssapi()

EZYRS64I GSS-API received a token of length *token_length*

Explanation: This is an orshd GSS-API informational message written to syslog. It is a progress message for debugging and indicates that the orshd server received the GSS-API token from the client.

token_length is the decimal length of the GSS-API token received from the orsh client.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshdgss

Procedure Name: do_gssapi()

EZYRS65I GSS-API sent token of length *token_length*

Explanation: This is an orshd GSS-API informational message written to syslog. It is a progress message for debugging and indicates that the orshd server sent the GSS-API token to the client.

token_length is the decimal length of the GSS-API token sent to the orsh client.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: rshdgss

Procedure Name: do_gssapi()

EZYRS66W -k start option not valid for IPv6 connection

Explanation: Kerberos version 5.0 or the GSSAPI is not supported for IPv6 connections.

System Action: Connection to the client is closed.

User or Operator Response: If tcp6 is specified in the inetd configuration file for the shell service, then start RSHD without the -k start option.

System Programmer Response: None.

Source Data Set: rshd.c

Procedure Name: main()

Chapter 11. EZYTxxxx messages

EZYTE01E Cannot translate current code page.

Explanation: This message is written to the client as well as the syslog file. An invalid return code was received from the routine `initxlate()`, using the default code page ISO8859-1. The connection ends. This could be the result of one of the following conditions:

- Invalid return from `iconv_open()` for building either the ASCII EBCDIC translation table or the EBCDIC to ASCII translation table.
- Invalid return from `iconv()` after opening the translate table.

System Action: Processing ends.

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

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

Source Data Set: `telnetd.c`

Procedure Name: `main`

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

Explanation: An invalid suboption has been specified in the `/etc/inetd.conf` file for `-a`, `-c`, `-D`, or `-X` option.

System Action: Processing ends.

User or Operator Response: None.

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

Source Data Set: `telnetd.c`

Procedure Name: `main`

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

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

System Action: Processing ends.

User or Operator Response: None.

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

Source Data Set: `telnetd.c`

Procedure Name: `main`

EZYTE04I *catgets description* rsn = *errnojr*

Explanation: This describes an error which occurred with the first attempt to retrieve a message from the message catalog. If this message appears, `otelneta` will use the default messages hard-coded within the software. Otherwise, `otelneta` will use the message catalog stored in the file `/usr/lib/nls/msg/C/tmmsgs.cat`.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None.

EZYTE05I • EZYTE09E

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main

EZYTE05I *Trace debug Debug diag* **keepalive** *keepaliv* **kludgeline***mode lmode* **hostinfo** *hostinfo* **Registered host**
registerd_host linemode always multi_proc multi_proc

Explanation: If -D report/all is specified, the above messages will be issued. They describe the parameter settings after the /etc/inetd.conf for telnet is read and processed. Trace flag corresponds to -t. Debug flag corresponds to -D. linemode flags corresponds to -l. Kludgeline mode corresponds to -k. Keepalives correspond to -n. hostinfo flags correspond to -h and Registered host - U. multi_proc corresponds to -m.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main

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

Explanation: This message is written to the client not the syslog file. It is informing the user that the parameters which were specified for telnet server are invalid.

System Action: Processing continues.

User or Operator Response: Check the /etc/inetd.conf for the parameters specified. Check the stderr file to determine which parameters were identified as invalid. The supported options will be printed in the syslog file. Correct the error and retry.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: usage

EZYTE08I **Usage: Supported options for telnetd:**

Explanation: This message displays a list of all supported options for otelnetd. See *z/OS Communications Server: IP Configuration Guide* for more information.

System Action: Processing ends.

User or Operator Response: Log in again after the system programmer corrects the invalid options.

System Programmer Response: Correct the invalid options in /etc/inetd.conf.

Source Data Set: telnetd.c

Procedure Name: usage

EZYTE09E **Telnet session is ending.**

Explanation: This message is written to the client as well as the syslog file. An error was found during processing the telnet server options.

System Action: Processing ends.

User or Operator Response: Correct the invalid option and re-execute the job.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: usage

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

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: terminaltypeok

EZYTE11I **doit:** **host_name** *host name* **doit:** **IP address** *IP_address* **doit:** **PORT** *port* **doit:** **host** *host_name*

Explanation: The following are variables that are set as a result of the gethostname(), gethostbyaddr(), getnameinfo(), or getaddrinfo().

- *host_name* is the name server name related to the IP address of the client.
- *IP_address* is the Internet address of the client.
- *port* is the port that the client is using.
- *host_name* is the MVS system being telnetted to.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit

EZYTE12E *userid* **Permission denied.**

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

System Action: Processing ends.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit

EZYTE13E **setgid failed** *description* **rsn = reason code**

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

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: Contact the system programmer.

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

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

Source Data Set: telnetd.c

EZYTE14E • EZYTE16I

Procedure Name: doit()

EZYTE14E initgroups failed *description* **rsn = errnojr**

Explanation: This message is written to the client as well as the syslog file. An initgroups subroutine failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: Contact the system programmer.

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

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

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE15E setuid failed *description* **rsn = errnojr**

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

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: Contact the systems programmer.

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

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

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE16I uid is: uid, gid is gid

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

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE17I interrupt()

Explanation: Entered the interrupt subroutine. Used for tracing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: interrupt()

EZYTE18I sendbrk()

Explanation: Entered the sendbrk subroutine. Used for tracing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: sendbrk()

EZYTE19I sendsusp()

Explanation: Entered the sendsusp subroutine. Used for tracing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: sendsusp()

EZYTE20I recv_ayt()

Explanation: Entered the recv_ayt subroutine. Used for tracing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: recv_ayt()

EZYTE21I doeof()

Explanation: Entered the doeof subroutine. Used for tracing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doeof()

EZYTE22I **herald()**

Explanation: Entered the herald subroutine. Used for tracing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE23E **herald: read error** *description* **rsn = errnojr**

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

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE24E **herald: open error** *description* **rsn = errnojr**

Explanation: An attempt was done to open /etc/banner in the herald. The open failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE25E **herald: malloc error** *description* **rsn = errnojr**

Explanation: An attempt was done to malloc/etc/banner in the herald. The malloc failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE26E **herald:** stat error *description* rsn = *errnojr*

Explanation: An attempt was done to stat /etc/banner in the herald. The stat failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: Correct the error indicated by *description* and *errnojr*. If an additional header is not stored in the /etc/banner file, specify -h on telnet configuration parameters to suppress the call to this routine for processing.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYTE27I **login:**

Explanation: This message is written to the client not the syslog file. A request from the server to the user to enter their userid.

System Action: Processing continues.

User or Operator Response: Specify the correct user ID for the system which you are telnet'ing to. Processing continues.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE28I **userid Password:**

Explanation: This message is written to the client not the syslog file. A request from the server to the user to enter their password.

System Action: Processing continues.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE29I **Starting new telnet session. catfd = catfd**

Explanation: Informational message identifying the first message issued by the server when diagnostics is specified. It also outputs the file descriptor associated with the message catalog. If this value is -1, then the user will be using the default messages hard-coded within the software. Otherwise, the user will be accessing the message catalog stored in the HFS /usr/lib/nls/msg/C/tnmsgs.cat.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main

EZYTE30E Invalid return code received from expired_pw().

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

System Action: Processing ends.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

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

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

System Action: Error

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE32W You entered an invalid login name or password.

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

System Action: Processing continues until after the third attempt, then it is ended.

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

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTE33W Password expired.

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

System Action: Processing continues.

User or Operator Response: The program will continue processing and request that the user specify a new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE34I Enter new password.

Explanation: This message is written to the client and not the syslog file. The return from `__passwd()` function indicates that the password entered has expired. A request for a new password is indicated.

System Action: Processing continues.

User or Operator Response: Enter new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE35I Re-enter new password.

Explanation: This message is written to the client and not the syslog file. Reenter the password again to check for typos on entering the expired password.

System Action: Processing continues.

User or Operator Response: Enter new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE36E You entered an invalid password.

Explanation: This message is written to the client as well as syslog file. A call was made to `__passwd()` to change the expired password. The new password specified resulted in errors.

System Action: Processing continues.

User or Operator Response: Check the *errno* and *errnojr* for more information. Enter new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE37E New passwords do not match.

Explanation: This message is written to the client as well as syslog file. The second password entered does not match the first password entered.

System Action: Processing continues.

User or Operator Response: reenter new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE38E Password too long.

Explanation: This message is written to the client as well as syslog file. The password specified is too long. Only a maximum of eight characters can be used for the password.

System Action: Processing continues.

User or Operator Response: Pick a new password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: expired_pw()

EZYTE39W Terminal type = DUMB. No full screen applications are supported

Explanation: This message is written to the syslog during diagnostics. Basically the terminal type specified during telnet negotiation was unsupported. Processing will continue but it will act as a dumb terminal. This means that all full screen applications will not work. Character-at-a-time raw mode applications such as **vi** will not work with a dumb terminal because they require full screen cursor support. Terminal types identified as 3270 will be changed to DUMB terminals.

System Action: Processing continues.

User or Operator Response: Do not use any full screen application.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: getterminaltype

EZYTE40E cleanup: waitpid failed. *description* rsn = errnojr

Explanation: The waitpid() provides a general interface that need to wait for certain child processes, that need resource utilization statistics accumulated by child process. A -1 was returned from the waitpid() for the child.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: The following conditions could be the result of an waitpid failure.

- The calling process has no existing unwaited-for child processes.
- The wait_status point to an illegal address.
- The call was interrupted by a caught signal or the signal did not have the SA_RESTART flag set.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE41I cleanup: child exit status = *wait status*

Explanation: The result of the waitpid specified during cleanup.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE44E Fork utmp error. *description* rsn = errnojr

Explanation: This message is written to the client as well as the syslog file. The parent process is trying to fork off a child process. It has not been able to do this.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

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

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

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

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE45E parent: utmp error = wait status

Explanation: The parent process is responding to the waitpid issued for the utmp processing. The parent could not cleanup the pid entry for the user ID specified.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Possibly the /etc/utmpx file had been updated already to remove the user ID associated with this pid number.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE46E EXECL: error description rsno = errnojr

Explanation: This message is written to the client as well as the syslog file. An error has occurred in trying to open the shell process to do the requested command.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

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

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

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

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTE47I • EZYTE51W

EZYTE47I **Int:** 20 bytes hex data 10 char ascii .

Explanation: This is hex/ascii representation of data which was inputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE48I **Ont:** 20 bytes hex data 10 char ascii .

Explanation: This is hex/ascii representation of data which was outputted from the telnet server to the client via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE49I **Ipt:** 20 bytes hex data 10 char ascii .

Explanation: This is hex/ascii representation of data which was inputted from the child to the telnet server via the master file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D ptydata or -D all. There may be an additional 4 bytes which are control data which will not be displayed with this message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE51W **terminaltypeok: Tgetent failure**

Explanation: The terminal type which was passed to the CURSES routine tgetent, did not match the supported types. Processing will attempt to negotiate a new terminal type. The supported terminal types can be found in /usr/lib/terminfo/ibm

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: terminaltypeok()

EZYTE52E **Couldn't resolve your address into a host name. IP address is** *IP address description* **rsn =** *errnojr*

Explanation: This message is written to the client as well as the syslog file. The function call to a `gethostbyaddr()` or `getnameinfo()` resulted in a failure. `gethostbyaddr()` or `getnameinfo()` returns a pointer to an object describing an Internet host referenced by address. This structure contains the information obtained from the name server. This message is followed by message EZYTE11I, which identifies the IP address which could not be found.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends. if -U specified. Otherwise, processing continues.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*. The following are possible reasons for failure:

- No such host is known.
- The local server did not receive a response from an authoritative server. Retry at some later time.
- Some unexpected server failure was encountered.
- The requested name is valid but does not have an IP address associated with this name.
- The local nameserver may be down.

Source Data Set: `telnetd.c`

Procedure Name: `doit()`

EZYTE54E **All network ports in use.**

Explanation: This message is written to the client as well as the syslog file. An attempt was made to identify ttys for the child process. An associated tty could not be obtained.

System Action: Processing ends.

User or Operator Response: Wait until some network ports are available.

System Programmer Response: Ensure that you have enough `/dev/ptypXXXX` and `/dev/ttypXXXX` for the number of users accessing your system. Examine the ports in use and ensure that none of them are zombies. Eliminate any zombies if found to free up some more ports.

Source Data Set: `telnetd.c`

Procedure Name: `doit()`

EZYTE55E **Getpty failed.**

Explanation: An attempt was made to identify ttys for the child process. An associated tty could not be obtained.

System Action: Processing ends.

User or Operator Response: Wait until some network ports are available.

System Programmer Response: Ensure that you have enough `/dev/ptypXXXX` and `/dev/ttypXXXX` for the number of users accessing your system. Examine the ports in use and ensure that none of them are zombies. Eliminate any zombies if found to free up some more ports.

Source Data Set: `telnetd.c`

Procedure Name: `doit()`

EZYTE56I **Yes**

Explanation: This message is written to the client not the syslog file. A request was issued for AYT by the user.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: recv_ayt()

EZYTE57I **TELNETD: netwrite *len* chars**

Explanation: The software issued data to be written to the client. This identifies the number of characters transmitted over the socket.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: recv_ayt

EZYTE59I **read_pw: Character ignored *char***

Explanation: Telnet was reading in information typed in by the user (usually user ID or password). There is an extraneous byte of information received after the carriage return (new line). This byte is ignored and processing is continued.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTE60I **cleanup(): child_pid = *child_pid***

Explanation: The child_pid specified is associated with the spawn issued at the start of this connection for the child.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE61E **parent: waitpid failed, status = *wait status***

Explanation: The waitpid() provides a general interface that need to wait for certain child processes, that need resource utilization statistics accumulated by child process. A -1 was returned from the waitpid() for the child.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: The following conditions could be the result of an waitpid failure.

- The calling process has no existing unwaited-for child processes.
- The wait_status point to an illegal address.

- The call was interrupted by a caught signal or the signal did not have the SA_RESTART flag set.

Source Data Set: telnetd.c

Procedure Name: cleanup()

EZYTE63I **Int:** 20 bytes hex data 10 char ascii .

Explanation: This is hex/ascii representation of data which was inputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to telnet negotiation data seen during processing of password and/or userid.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTE64I **Int:** 20 bytes hex data 10 char ascii .

Explanation: This is hex/ascii representation of data which was inputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to input data received for password/user ID which exceeds the length of the buffer for password/user ID (8 bytes). This data will be ignored for further processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTE65I **PROTOCOL:** send IAC Data Mark.

Explanation: A Data mark was sent to the client by the server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE66I **PROTOCOL:** lmodetype = lmodetype linemode = linemode useline useline

Explanation:

lmodetype signifies whether the client handle real linemode or if use of kludgeline mode is needed. It will be set to one of the following:

- 0x04 REAL-LINEMODE - use the linemode option
- 0x03 KLUDGE-OK
- 0x02 NO-AUTOKLUDGE
- 0x01 KLUDGE-LINEMODE - use kludge linemode
- 0x00 NO-LINEMODE - client is ignorant of linemode

linemode is true if linemode is currently on.

EZYTE67I • EZYTE70W

uselinemode is the state that we wish to be in.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE67I S(nfd): sockfd..ibits=ibits obits=obits ebits=ebits S(nfd): pty...ibits=ibits obits=obits ebits=ebits

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the read, write and exceptions on either the masterfd (pty) or the sockfd. They are associated with what will happen within the protocol function. They determine if reading/writing of the master file descriptor or the socket file descriptor will occur. They will also determine if there is an exception outstanding on a particular pty which needs to be addressed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE68I Ept: #bytes = cc pkcontrol(cntl) cntl

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the exceptions on the pty. An exception condition was identified which needs to be processed. This identifies the number of bytes read in for the exception and what the actual exception condition was which needs to be handled.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE69I PROTOCOL: cntl = cntl

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the exceptions on the pty. The resulting exception condition after it has been processed with out-of-band data.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE70W PROTOCOL: Found an unknown exception.

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the exceptions on the pty. An exception was read in which we are not processing currently. It is ignored. The ones currently handled are:

- FLUSHWRITE
- DOSTOP
- NOSTOP

- CHCP
- IOCTL

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE71I SYNCHing is turned on

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the exceptions on the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE72I PROTOCOL: netread *ncc* chars.

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. Identifies the number of bytes read in from the client to be either sent on to the child or processed by the server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE73I PROTOCOL: EOF on socket.

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the exceptions on the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE74I PROTOCOL: EOF on master tty.

Explanation: These messages are only issued if -t is specified in /etc/inetd.conf. They correlate to the conditions of the exceptions on the pty. An EOF was received on the master tty. This results in the cleanup routine being called and the session eventually terminated.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

EZYTE78I • EZYTE81E

Procedure Name: telnet()

EZYTE78I PROTOCOL: ptyread *pcc* chars

Explanation: These messages are only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the number of bytes read in from the child. They will be sent on to the client after processing by the server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE79E PROTOCOL: select mask too small, increase FD_SETSIZE *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file. Select uses bit masks of file descriptors in longs. FD_SETSIZE is defaulted to 2048.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Increase the FD_SETSIZE to an appropriate number and rebuild the server.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE80W PROTOCOL: select *description* rsn = *errnojr*

Explanation: This message is written to the syslog file when the select() function Call has terminated due to a timeout value being exceeded. The telnet session is ended.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System Action: Session is cancelled due to timeout.

User or Operator Response: None.

System Programmer Response: Check the -c value in the /etc/inetd.conf file to verify that the timeout value is appropriate.

- The select() timed out.
- The telnet session is ended due to the select() timeout.
- Check the -c select timeout value in the /etc/inetd.conf file.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE81E PROTOCOL: ioctl net FIONBIO. *description* rsn = *errnojr*

Explanation: An unexpected value was returned from the ioctl.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*. The following are some possible explanations of the failure:

- The file descriptor specified is invalid.
- The file descriptor specified is not associated with a character special device.
- The specified request does not apply to the kind of object that the socket file descriptor references.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE82E PROTOCOL: error fcntl masterfd FIONBIO. *description* *rsn = errnojr*

Explanation: An unexpected value was returned from the ioctl.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*. The following are some possible explanations of the failure:

- The file descriptor specified is invalid.
- The file descriptor specified is not associated with a character special device.
- The specified request does not apply to the kind of object that the socket file descriptor references.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE84I PROTOCOL: SIMULATING receive

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE85I SYNCHing = SYNCHing

Explanation: An unexpected value was returned from the ioctl.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTE86W TELNETD: setitimer *description* **rsn = errnojr**

Explanation: This message is written to the syslog file when the setitimer expires due to the timeout value being exceeded before the user had logged in. The telnet session is ended.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Session is cancelled due to timeout.

User or Operator Response: None.

System Programmer Response: Check the -c value in the /etc/inetd.conf file to verify that the timeout value is appropriate.

- The user timed out before logging in.
- The telnet session is ended due to the user timeout.
- Check the -c timeout value in the /etc/inetd.conf file.

Source Data Set: telnetd.c

Procedure Name: telnet()

EZYTE87W TELNETD: sigaction() failed - *description* **rsn = errnojr**

Explanation: This message is written to the syslog file when the sigaction() has failed. This will cause the user timeout value before login (-c option) not to be set. The telnet session continues without the timeout set.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Signal handler for SIGALRM not set up, user timeout before login will not be set.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: telnetd.c

Procedure Name: telnet()

EZYTO01I Int: 20 bytes hex data 10 char ascii .

Explanation: This is hex/ascii representation of data which was inputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to input data received for password/user ID processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTO02I Ont: 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was outputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to data received for password/user ID processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTO03E Incoming session is not from a registered host.

Explanation: The server has been configured with a -U option which stipulates that only registered hosts will be accepted. The telnet session currently being processed is not registered and will be rejected.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Ensure that the incoming address is specified in the appropriate name server or etc/host file.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTO04I lusername = userid

Explanation: During processing of the userid/password, this message is issued if diagnostics is turned on. It will print out the user ID which was entered.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: verify_password()

EZYTO05I Initial EBCDIC codepage = codepage, ascii codepage = codepage

Explanation: If diagnostics are on, this will print out what the initial code page settings are.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: main()

EZYTO06I Int: 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was outputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to data received for banner processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYT007I **Ont:** 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was outputted to the client from the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to data received for banner processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: herald()

EZYT008E **Password too long.**

Explanation: This message is written to the client as well as syslog file. The password specified is too long. Only a maximum of eight characters can be used for the password.

System Action: Processing continues.

User or Operator Response: Specify a correct password.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYT009I **options(entry) = value .**

Explanation: This message is written to the syslog file. It identifies which options have been negotiated prior to the exec() taking place in the telnet server. The option array entry corresponds to a telnet option. The resulting value represents what was negotiated. If the value is 3 it correlates with the client has agreed to perform that function. If it is 12, then the server has agreed to perform this function.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYT010E **Terminaltype is not recognized. save_ttype**

Explanation: The client responded to a DO TERMINAL TYPE, with a WILL TERMINAL TYPE. It then proceeded to send via subnegotiation, terminal types which are not supported by this server. As a valid terminal type was not received the session is ended.

System Action: Processing ends

User or Operator Response: Ensure that the client you are using can support terminal types which are supported by the curses function of tgetent(). At this time, this does not include 3270. Therefore, if your client is attempting to emulate a 3270 data stream the connection will not be accepted by this server.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: doit()

EZYTO11I **Int:** 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was outputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all. It corresponds to data received for telnet subnegotiation which was deferred until the tty was active.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: undo_chars()

EZYTO12E **Logon user name too long.**

Explanation: This message is written to the client as well as syslog file. The logon user name specified is too long. Only a maximum of eight characters can be used for the userid.

System Action: Processing continues.

User or Operator Response: Specify a correct userid.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: read_pw()

EZYTS02I **STATE:telrcv: send IAC c**

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation SENT during the execution of the job.

System Action: Telnet Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: telrcv

EZYTS03E **STATE:telrcv: panic state = state**

Explanation: An IAC command is found during telnet negotiation and is followed by an unrecognized command option.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Issue traces to determine what was attempting to be negotiated as telnet control data. Trace to issue from server is '-D options'.

Source Data Set: state.c

Procedure Name: telrcv

EZYTS04I **STATE:send_do: send option**

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a request to issue a DO option.

System Action: Processing continues.

EZYTS05I • EZYTS08I

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_do

EZYTS05I STATE:willoption: receive WILL *option*

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically the receipt of WILL request.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: willoption

EZYTS06I STATE: willoption: set to kludge ok

Explanation: This is an informational message which is only issued if the user has specified -t in the /etc/inetd.conf file. It informs the user that kludgelinemode is operational.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: willoption

EZYTS07I STATE:send_dont: send DON'T *option*

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a request to issue a DONT option.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_dont

EZYTS08I STATE:wontoption: receive WON'T *option*

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a WONT was received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: wontoption

EZYTS09I STATE:send_will: send WILL option

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a WILL was sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_will

EZYTS10I STATE:doooption: receive DO option

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a DO was received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: dooption

EZYTS11I STATE:send_wont: send WONT option

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a WONT was sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: send_wont

EZYTS12I STATE:dontoption: receive DONT option

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a DONT was received.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: dontoption

EZYTS13I ENVIRON VALUE and VAR are reversed!

Explanation: This is an informational message which is only issued if the user has specified -D options in the /etc/inetd.conf file. This is issued during suboption negotiation. To be interoperable we need to determine if the VALUE and VAR values are reversed. If the first recognized character is a VAR or VALUE, then that will tell what type of client it is. If the first recognized character is a USERVAR, then we continue scanning the suboption looking

EZYTS14I • EZYTS17I

for two consecutive VAR or VALUE fields. We should not get two consecutive VALUE fields. If a client has sent a well-formed option then the number of VALUES received should always be less than or equal to the number of VARS and USERVARs received. If not then the client has reversed the definitions.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: suboption

EZYTS14I **Opt:** 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was Outputted from the client to the child via the master file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D ptydata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: telrcv()

EZYTS15I **STATE:doooption:deferred receive DO** option

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a DO was received prior to the creation of the tty which may require special tty processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: dooption

EZYTS16I **STATE:willoption:deferred receive WILL** option

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. It informs the user of internal telnet negotiation occurring during the execution of the job. Specifically a response that a WILL was received prior to the creation of the tty which may require special tty processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: willoption

EZYTS17I **STATE:Defer subotopn negotiation.**

Explanation: This is an informational message that is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. Subnegotiation data was received that must be deferred until the tty is active.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: suboption()

EZYTS18I STATE:Process deferred subotopn negotiation.

Explanation: This is an informational message which is only issued if the user has specified -D options/report/all in the /etc/inetd.conf file. Subnegotiation data was received that must be deferred until the tty is active. Now we can process this information.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: state.c

Procedure Name: suboption()

EZYTU01E UTILITY:Read from ttloop. *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file. The number of bytes returned from a read on the socketfd (client) is invalid. A negative value was received. The connection is ended.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

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

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop

EZYTU02E UTILITY:Read from ttloop. Peer died. *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file. The number of bytes returned from a read on the socketfd (client) is invalid. A value of zero was received. The connection is ended.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

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

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop

EZYTU03I **UTILITY:** `ttloop read ncc chars`.

Explanation: This message is issued if `-D report` or `-D ptydata`, is specified. It will write the number of characters that were read in from the socketfd, (client's socket) to be processed. It will be followed by hex data, preceded by the tag **Int**. This signifies a hex and ascii representation of input data coming into the telnet server from the network.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: `utility.c`

Procedure Name: `ttloop`

EZYTU04E **Telnetd: UTILITY:** **Can not translate current code page.**

Explanation: This message is written to the client as well as the syslog file. An error was found while processing the `initxlate()` routine. The function will terminate the connection.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check for additional messages issued.

Source Data Set: `utility.c`

Procedure Name: `ReturnToDefault`

EZYTU05E **UTILITY:** `__tcsetcp. description rsn = errnojr`

Explanation: This message is written to the client as well as the syslog file. An error was found while processing the default code pages.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: `utility.c`

Procedure Name: `ReturnToDefault`

EZYTU06E **Error using code pages *newNames.__tcp_toname* and *newNames.__tcp_fromname* - returning to default code pages.**

Explanation: This message will be issued to the client and only to syslog file, if `debug_mode` is specified. The current code pages are not working correctly. Telnet is returning to the default code pages. If `-D netdata` is specified, this message will be followed by the hex and ascii translation of this message preceded by the tag 'Ont'.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check the installation of the code pages. If requested code page is not valid, generate the appropriate codepage. Otherwise, if the option is valid, check to see if codepage is still good.

Source Data Set: `utility.c`

Procedure Name: `ReturnToDefault`

EZYTU07E **UTILITY:** `__tcgetcp`. *description* **rsn** = *errnojr*

Explanation: This message is written to the client as well as the syslog file. An error was found while changing the code pages.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The connection is ended.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: `utility.c`

Procedure Name: `change_translate`

EZYTU08I **Telnetd: UTILITY: Change to binary mode.**

Explanation: The tty has been switched to binary mode.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: `utility.c`

Procedure Name: `change_translate`

EZYTU09I **UTILITY: Change to SingleByte pages** *newNames.__tcp_toname* *newNames.__tcp_fromname*

Explanation: This is a single byte code page. A call to `initxlate()` will be made to setup the new code pages.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: `utility.c`

Procedure Name: `change_translate`

EZYTU10I **UTILITY: Change to MultiByte pages** *newNames.__tcp_toname* *newNames.__tcp_fromname*

Explanation: This is a multi-byte code page. A call to `doMultiByte()` will be made to setup the new code pages.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: `utility.c`

Procedure Name: `doMultiByte`

EZYTU11E **UTILITY: Unexpected input string** *description* **rsn** = *errnojr*

Explanation: This message is written to the client as well as the syslog file. The `errno` identified from `iconv` (on the `ascii` to `EBCDIC` converter) was `EINVAL`. `EINVAL` should only be encountered when the last character in the input buffer is incomplete. This did not occur.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in

EZYTU12E • EZYTU14I

the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: A2Emultiybyte_translate

EZYTU12E UTILITY: Unexpected iconv error *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file. The errno identified from iconv (on the ascii to EBCDIC converter) is not currently checked for. Unexpected error detected.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: A2Emultiybyte_translate

EZYTU13E UTILITY: stilloob: select *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file. A check on the file descriptor was made to determine if out of band data existed. An invalid value was returned. The value was less than zero.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: stilloob.

EZYTU14I UTILITY: netwrite *n* chars.

Explanation: This message is issued if -D report or -D ptydata, is specified. It will write the number of characters which were read in from the sockfd, (client's socket) to be processed. It will be followed by hex data, preceded by the tag "Ont". This signifies a hex and ascii representation of Output data coming from the Telnet server to the net.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: netflush

EZYTU15I **UTILITY:** `ptywrite n chars.`

Explanation: This message is issued if -D report or -D ptydata, is specified. It will write the number of characters which were read in from the socketfd, (client's socket) to be processed. It will be followed by hex data, preceded by the tag "Opt". This signifies a hex and ascii representation of Output data coming from the Telnet server to the child.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ptyflush

EZYTU16I **UTILITY:** `Write of masterfd. description rsn = errnojr`

Explanation: This message is issued if -D report or -t, is specified. A negative value was received of the number of bytes written to the masterfd. Processing continues and control is returned to the calling routine.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ptyflush

EZYTU17I **UTILITY:** `direction suboption sub_option`

Explanation: A telnet negotiated suboption was either received or sent by the client.

direction is either send or receive.

sub_option is the Telnet negotiation suboption.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: printsub

EZYTU18I **UTILITY:** `write/send from NETFLUSH`

Explanation: A telnet negotiated suboption was either received or sent by the client. The remaining portion of this command identifies what the suboption was.

System Action: Processing continues.

User or Operator Response: None. Debug trace data.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: printsub

EZYTU19E **UTILITY: Unexpected iconv open error** *description* **rsn = errnojr**

Explanation: The iconv_open failed for ascii to EBCDIC converter. The errno identified from iconv (on the ascii to EBCDIC converter) is not currently checked for. Unexpected error detected.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: doMultiByte()

EZYTU20I **Int:** 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was inputted from the client to the telnet server via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ttloop()

EZYTU21I **Ont:** 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was outputted from the telnet server to the client via the socket file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D netdata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: netflush()

EZYTU22I **Opt:** 20 bytes hex data 10 char ascii.

Explanation: This is hex/ascii representation of data which was Outputted from the client to the child via the master file descriptor. This is only seen during tracing of the server. The /etc/inetd.conf must have specified -D ptydata or -D all.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: ptyflush()

EZYTU23E UTILITY: Unexpected iconv error *description* **rsn = errnojr**

Explanation: This message is written to the client as well as the syslog file. The errno identified from iconv (on the EBCDIC to ascii converter) is not currently checked for. Unexpected error detected.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultiybyte_translate

EZYTU24E UTILITY: Unexpected iconv error. *description* **rsn = errnojr**

Explanation: This message is written to the client as well as the syslog file. An unexpected error condition was received from the iconv call.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultiybyte_translate

EZYTU25E UTILITY: Unexpected iconv error *description* **rsn = errnojr**

Explanation: Building an ascii to EBCDIC translate table from the default tables. An error was produced during processing. Check the errno and errnojr for corrective action.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: initxlate()

EZYTU26E UTILITY: Unexpected iconv error *description* **rsn = errnojr**

Explanation: Building an EBCDIC to ascii translate table from the default tables. An error was produced during processing. Check the errno and errnojr for corrective action.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in

EZYTU27E • EZYTU29E

the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: initxlate()

EZYTU27E UTILITY: Unexpected iconv open error *description* **rsn = errnojr**

Explanation: The iconv_open failed for EBCDIC to ascii converter. The errno identified from iconv (on the EBCDIC to ascii converter) is not currently checked for. Unexpected error detected.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: doMultiByte()

EZYTU28E UTILITY: Unexpected iconv open error *description* **rsn = errnojr**

Explanation: The iconv_open failed for ascii to EBCDIC converter. The errno identified from iconv (on the ascii to EBCDIC converter) is not currently checked for. Unexpected error detected.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: initxlate()

EZYTU29E UTILITY: ascii translation error *Xbuf->begin_translate*

Explanation: The iconv failed for ascii to EBCDIC converter.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Ensure that the converter table being specified is valid and exists.

Source Data Set: utility.c

Procedure Name: A2Emultibyte_translate()

EZYTU30I **UTILITY:** A2E Begin_write len=*bytes_translated*,data=*begin_w*

Explanation: Tracing information inserted to help debug conversion errors. It prints out both the length of the characters being converted as well as the character set. It is invoked using -t option in the setup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: A2Emultibyte_translate()

EZYTU31I **UTILITY:**E2A Begin_translate len=*inlen*,data=*begin_w*

Explanation: The iconv failed for EBCDIC to ascii converter. Tracing information inserted to help debug conversion errors. It prints out both the length of the characters being converted as well as the character set. It is invoked using -t option in the setup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultibyte_translate()

EZYTU32E **UTILITY:**EBCDIC translation error *bad char*

Explanation: Tracing information inserted to help debug conversion errors. It prints out both the length of the characters being converted as well as the character set. It is invoked using -t option in the setup parameters.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: E2Amultibyte_translate()

EZYTU33E **UTILITY:**Termination loop detected - telnet exiting

Explanation: A loop was detected during termination while processing a fatal error. The telnet server will terminate immediately after printing this message.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: utility.c

Procedure Name: fatal()

EZYTU34I **id id pri pri call failing call code errno reason errnojr h_errno h_errno**

Explanation: A system or library call failed.

id This identifies the location within the source code that recorded the error.

pri This is the value passed as the first parameter to the syslog() function. See the description of the syslog() function in *z/OS XL C/C++ Run-Time Library Reference* for more information.

failing call

This is the system or library call which failed. See the description of this system or library call in *z/OS XL C/C++ Run-Time Library Reference* for more information.

errno This is the hexadecimal UNIX System Services Return Code. These return codes are listed and described in the Return Codes (Errnos) chapter in *z/OS UNIX System Services Messages and Codes*. If there is no *errno*, this field will display as **N/A**.

errnojr This is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojrs) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed. If there is no *errnojr*, this field will display as **N/A**.

h_errno This error code is no longer applicable and will always display as **N/A**.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

EZYTU35I Error messages will not be saved in /tmp/telnetd.stderr, reason *reason*

Explanation: Messages written to stderr in the otelnetd process are usually stored in /tmp/telnetd.stderr. These messages will be discarded for one of the following reasons:

- 1 /tmp/telnetd.stderr could not be opened for append or created.
- 2 A system or library call failed while processing /tmp/telnetd.stderr. See a prior EZYTU34I record for an explanation.
- 3 /tmp/telnetd.stderr exists but is not a regular file. /tmp/telnetd.stderr must be a regular file. Other types of files, such as symbolic links, cannot be used because of possible security exposures.
- 4 /tmp/telnetd.stderr was re-created by another process while being opened and checked by this otelnetd process.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

EZYTU36I *code userid local_IP local_port remote_IP remote_port client_hostname*

Explanation: This trace record is written to the syslogd facility **auth** when the **-D login** or **-D all** parameter is specified on the otelnetd command line. It provides a one-line summary of login and logout activity.

code describes which activity is being recorded:

- L** The user successfully logged on to the system.
- O** The user logged off.
- U** The user provided a user ID which was incorrect.
- P** The user provided a password which was incorrect.
- C** The user did not successfully change the password.

userid is the user ID specified by the user when logging on.

local_IP is the destination IP address specified by the user when connecting to the telnet server.

local_port is the destination TCP port specified by the user when connecting to the telnet server.

remote_IP is the IP address of the host used by the client to connect to the telnet server.

remote_port is the TCP port assigned to the telnet client.

client_hostname is the DNS name of the host used by the client to connect to the telnet server, if that information is available. If not available, the *client_hostname* field in the trace record will contain a hyphen.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

EZYTY01E GETPTY: Out of ptys. *description* **rsn = errnojr**

Explanation: This message is written to the client as well as the syslog file. All of the /dev/ptypXXX are currently in use with other sessions or processes.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: User has to wait until some /dev/ptypXXX are freed up for use.

System Programmer Response: Check to determine if there are any ghosts or zombies hanging around that can be released.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTY02I GETPTY: open of /dev/pty

Explanation: These messages are only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: protocol.h

Procedure Name: telnet()

EZYTY03I gotpty: ioctl TIOCSWINSIZ

Explanation: These messages are only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY04E GETPTY: open error on *line description* **rsn = errnojr**

Explanation: This message is written to the client as well as the syslog file. A problem occurred trying to open the corresponding /dev/typXXX file.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

EZTYTY05I • EZTYTY07E

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

- Access is denied.
- The process is busy.
- The open was interrupted by a signal.
- The system has reached the maximum number of file descriptors it can have open.
- Permission to open is denied for one of the following reasons.
 - The user who opened the master tty is not the same user associated with the slave tty.
 - Internal security error.
 - Different path name was specified for the slave than earlier opens.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZTYTY05I GETPTY: slave fd = *slavefd*, masterfd = *masterfd*

Explanation: These messages are only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZTYTY06E gotpty: __tcsetattr *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZTYTY07E gotpty: __tcsetcp *description* rsn = *errnojr*

Explanation: This message is written to the client as well as the syslog file. Issued a tcsetcp to inform the tty of the code pages that we are using to translate ascii to EBCDIC and the reverse. The __tcsetcp failed.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the z/OS UNIX System Services Messages and Codes, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZYTY08I **argv_fsum**(*argument_number*) = *argument_value*

Explanation: These are the parameters that are passed to the spawned process that creates the child. This message is issued once for each argument. A few of these variables are set by the server.

- argv_fsum(0) = argument name
- argv_fsum(4) = master file descriptor
- argv_fsum(5) = slave file descriptor
- argv_fsum(8) = Debug tracing variable
- argv_fsum(9) = Debug tracing variable
- argv_fsum(11) = terminal type

argument_number is the argument number. It will be in the range 0–16.

argument_value is the value assigned to that argument.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: login_tty()

EZYTY09I **login_tty: spawnp fsumoclp** *child_pid*

Explanation: The pid number associated with the fork() for FSUMOCLO for cleanup.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: login_tty()

EZYTY10E **login_tty: spawnp error** *child_pid* *description* **rsn** = *errnojr*

Explanation: This message is written to the client as well as the syslog file. The pid number associated with the fork() for FSUMOCLO for cleanup.

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Codes (Errnojr) chapter of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: telnetd.c

Procedure Name: login_tty()

EZTYTY11I GETPTY: stat of /dev/ptyp

Explanation: This message is only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZTYTY12I ioctl masterfd TIOCEXT

Explanation: This message is only issued if -D report or -D netdata specified in /etc/inetd.conf. They correlate to the conditions of the writing for the pty.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sys_term.h

Procedure Name: tty_setlinemode()

EZTYTY13E login_tty failed.

Explanation: This message is written to the client as well as the syslog file. The login_tty() routine was called which sets up and spawns the tty. Errors were found during the processing and a return code of -1 was sent back to the caller.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Check for further messages in the log to explain reason for the failure.

Source Data Set: sys_term.h

Procedure Name: gotpty()

EZTYTY14E tcgetattr() failed on master tty

Explanation: A tcgetattr() was issued on the master tty, which is in fact an ioctl(). It returned with an unexpected error.

System Action: Processing ends.

User or Operator Response: None.

System Programmer Response: Correct the error indicated by *description* and *errnojr*.

Source Data Set: sys_term.h

Procedure Name: init_termbuf()

Chapter 12. EZYXxxxx messages

EZYXM01W The arrow direction is not correct.

Explanation: XmNarrowDirection resource in XmArrowButton widget class. XmNarrowDirection resource in XmArrowButtonGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ArrowB.c, ArrowBG.c

EZYXM02W Incorrect resize policy.

Explanation: XmNresizePolicy resource in XmBulletinBoard widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM03W Incorrect dialog style.

Explanation: XmNdialogStyle resource in XmBulletinBoard widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM04W Incorrect shadow type.

Explanation: XmNshadowType resource in XmBulletinBoard widget class. XmNshadowType resource in XmDrawnButton widget class. XmNshadowType resource in XmFrame widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM05W Null font list (no vendor shell default).

Explanation: XmNbuttonFontList resource, XmNlabelFontList resource, or XmNtextFontList resource in XmBulletinBoard widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BulletinB.c

EZYXM06W Dialog style must be XmDIALOG_MODELESS

Explanation: The Initialize function or the SetValues function detected that the bulletin board dialog style was not set as XmDIALOG_MODELESS, when XmDIALOG_MODELESS is required.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library

Procedure Name: BulletinB.c

EZYXM11W XmCascadeButton must have correct type of XmRowColumnWidgetClass parent.

Explanation: XmCascadeButton widget class. XmCascasdeButtonGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeBG.c

EZYXM12W Only XmMENU_PULLDOWN XmRowColumnWidgets can be submenus.

Explanation: XmCascadeButton widget class. XmCascadeButtonGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, CascadeBG.c

EZYXM13W MapDelay must be >= 0.

Explanation: XmNmappingDelay resource in XmCascadeButton widget class. XmNmappingDelay resource in XmCascadeButtonGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, CascadeBG.c

EZYXM14W **XmCascadeButtonGadget must have XmRowColumnWidgetClass parent with XmNrowColumnType XmMENU_PULLDOWN, XmMENU_POPUP, XmMENU_BAR or XmMENU_OPTION.**

Explanation: XmCascadeButton widget class. XmCascadeButtonGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c

EZYXM15W **XtGrabPointer failed.**

Explanation: XGrabPointer function. XtGrabPointer function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, MenuUtil.c, TrackLoc.c

EZYXM16W **XtGrabKeyboard failed.**

Explanation: XGrabKeyboard function. XtGrabKeyboard function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MenuUtil.c, TextF.c, TextIn.c

EZYXM17W **Only XmRowColumn widgets of type XmMENU_PULLDOWN can be submenus.**

Explanation: The Initialize function or the SetValues function detected that a submenu in the processing Widget is not of type XmMENU_PULLDOWN. The submenu Widget is reset to NULL.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: CascadeB.c, CascadeBG.c

EZYXM21W The dialog type must be XmDIALOG_COMMAND.

Explanation: XmCommand widget class. XmNdialogType resource in XmSelectionBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM22W Invalid child type. The Command widget does not have this child.

Explanation: XmCommandGetChild function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM23W Invalid XmString, check for invalid charset.

Explanation: XmCommandAppendValue function. XmCommandSetValue function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM24W NULL or empty string passed in to CommandAppendValue.

Explanation: XmCommandAppendValue function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM25W XmNmustMatch is always False for a Command widget.

Explanation: XmCommand widget class. XmNmustMatch resource in XmSelectionBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM26W XmNhistoryMaxItems must be a positive integer greater than zero.

Explanation: The Initialize function or the SetValues function detected that the XmNhistoryMaxItems for the XmCommand Widget has a value less than zero. The XmNhistoryMaxItems is reset to the initialized value or 100.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Command.c

EZYXM31W Must call XmClipboardStartCopy() before XmClipboardCopy().

Explanation: XmClipboardCopy function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM32W Must call XmClipboardStartCopy() before XmClipboardEndCopy().

Explanation: XmClipboardEndCopy function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM33W Too many formats in XmClipboardCopy().

Explanation: XmClipboardCopy function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c Do not translate rectObj

EZYXM35W Incorrect data type.

Explanation: The ClipboardFindItem function detected that the data type passed to the Clipboard does not match the data type of the Clipboard Pointer.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

EZYXM36W • EZYXM37W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM36W ClipboardCorrupt

Explanation: One of the following functions detected a clipboard error, and called the ClipboardError routine with key CLIPBOARD_CORRUPT:

- ClipboardFindFormat
- ClipboardDeleteFormat
- ClipboardDeleteFormats
- ClipboardDeleteItemLabel
- ClipboardIsMarkedForDelete
- ClipboardMarkItem
- ClipboardDataIsReady
- XmClipboardEndCopy
- XmClipboardCopyByName
- XmClipboardUndoCopy
- ClipboardRetrieve

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM37W Internal error - corrupt data structure.

Explanation: An internal application error was detected. One of the following functions detected that either the ClipboardDataItem or ClipboardFormatItem did not contain the required data structure:

- ClipboardFindFormat
- ClipboardDeleteFormat
- ClipboardDeleteFormats
- ClipboardDeleteItemLabel
- ClipboardIsMarkedForDelete
- ClipboardMarkItem
- ClipboardDataIsReady
- XmClipboardEndCopy
- XmClipboardCopyByName
- XmClipboardUndoCopy
- ClipboardRetrieve

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM38W Registered format length must be 8, 16, or 32

Explanation: XmClipboardRegisterFormat function detected that the format_length variable did not contain an allowed value.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM39W Registered format name must not be NULL.

Explanation: The XmClipboardRegisterFormat function detected that the format_name variable did not contain an allowed value.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: CutPaste.c

EZYXM41W DialogShell widget only supports one rectObj child.

Explanation: XmDialogShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DialogS.c

EZYXM42W gadgets aren't allowed in shell.

Explanation: XmDialogShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DialogS.c

EZYXM43W DialogShell widget supports only one RectObj child

Explanation: The InsertChild function detected that the GetRectObjKid function failed because the input child is a CoreClass object, instead of the required RectObj object.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

EZYXM51W • EZYXM62W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DialogS.c

EZYXM51W Margin width or height cannot be negative.

Explanation: XmDrawingArea widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DrawingA.c

EZYXM52W Incorrect resize policy.

Explanation: XmNresizePolicy resource in XmDrawingArea widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DrawingA.c

EZYXM61W Fraction base cannot be zero.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM62W Incorrect form attachment type.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM63W Cannot set constraints for non-resizable widget.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM64W Attachment widget must not be null.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM65W Circular dependency in Form children.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM66W Edge attached to a widget but no widget specified.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM67W Bailed out of edge synchronization after 10,000 iterations. Check for contradictory constraints on the children of this form.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

EZYXM68W • EZYXM81W

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM68W Attachment widget must be have same parent as widget.

Explanation: XmForm widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Form.c

EZYXM72W Only one child should be inserted in a frame.

Explanation: XmFrame widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Frame.c

EZYXM73W Invalid margin width.

Explanation: XmFrame widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Frame.c

EZYXM74W Invalid margin height.

Explanation: XmFrame widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Frame.c

EZYXM81W Invalid highlight thickness.

Explanation: XmGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM82W The unit type is incorrect.

Explanation: XmNunitType resource in XmGadget widget class. XmNunitType resource in XmManager widget class. XmNunitType resource in XmPrimitive widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM83W Invalid shadow thickness.

Explanation: XmGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM84W Cannot set pixmap resource to unspecified.

Explanation: XmNtopShadowPixmap resource in XmPrimitive widget class. XmNbottomShadowPixmap resource in XmPrimitive widget class. XmNhighlightShadowPixmap resource in XmPrimitive widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM85W Cannot change XmNlayoutDirection after initialization.

Explanation: The SetValues function detected that the XmNlayoutDirection value was changed after initialization. The XmNlayoutDirection value is reset to the initialized value.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Gadget.c

EZYXM91W Invalid XmNlabelType.

Explanation: XmNlabelType resource in XmLabel widget class. XmNlabelType resource in XmLabelGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM92W Invalid value in XmNalignment.

Explanation: XmNalignment resource in XmLabel widget class. XmNalignment resource in XmLabelGadget widget class. XmNstringDirection resource in XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM93W Invalid value in XmNstringDirection.

Explanation: XmNstringDirection resource in XmLabel widget class. XmNstringDirection resource in XmLabelGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM94W Invalid XmNlabelString - must be a compound string.

Explanation: XmNlabelString resource in XmLabel widget class. XmNlabelString resource in XmLabelGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXM95W Invalid XmNacceleratorText - must be a compound string.

Explanation: XmNacceleratorText resource in XmLabel widget class. XmNacceleratorText resource in XmLabelGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Label.c

EZYXN00W List must have at least one visible item.

Explanation: XmNvisibleItemCount resource in XmList widget class. When changed, XmNvisibleItemCount must be at least 1.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN01W Invalid Selection Policy.

Explanation: XmNselectionPolicy resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN02W Invalid Size Policy.

Explanation: XmNlistSizePolicy resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN03W Invalid ScrollBar Display Policy.

Explanation: XmNscrollbarDisplayPolicy resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN04W Invalid String Direction.

Explanation: XmNstringDirection resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN05W Cannot change size policy after initialization.

Explanation: XmNlistSizePolicy resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN06W Must set item count to non-negative value.

Explanation: XmNitemCount resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN07W NULL font in SetValues ignored.

Explanation: XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN08W Invalid item(s) to delete.

Explanation: XmNitemCount resource in XmList widget class. XmListDeleteItem function. XmListDeleteItems function. XmListDeletePos function. XmListDeleteItemsPos function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN09W No Horizontal Scrollbar to set.

Explanation: XmListSetHorizPos function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN10W Invalid Margin setting.

Explanation: XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN11W Invalid Spacing Value.

Explanation: XmNlistSpacing resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN12W Cannot set items to NULL with non-zero item count.

Explanation: XmNitemCount resource in XmList widget class. XmNitems resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN13W Must set selected item count to non-negative value.

Explanation: XmNselectedItemCount resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

EZYXN14W • EZYXN17W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN14W Cannot set selected items to NULL with non-zero item count.

Explanation: XmNselectedItemCount resource in XmList widget class. XmNselectedItems resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN15W Cannot set top position less than 1.

Explanation: XmNtopItemPosition resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN16W XmNitems and XmNitemCount mismatch!

Explanation: XmNitemCount resource in XmList widget class. XmNitems resource in XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN17W Cannot leave add mode in multiple selection.

Explanation: XmList widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN18W XmNselectedPositionCount must not be negative.

Explanation: The SetValues function detected that the XmNselectedPositionCount was either negative or zero, when a positive value was required.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN19W Cannot set XmNselectedPosition to NULL when XmNSelectedPositionCount is positive.

Explanation: The SetValues function detected an error with the XmNselectedPosition and XmNSelectedPositionCount. The selected position is set to NULL, but the position count contains a positive value.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: List.c

EZYXN21W The Menu Bar cannot be changed to NULL.

Explanation: XmNMenuBar resource in XmMainWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MainW.c

EZYXN22W The Command Window cannot be changed to NULL.

Explanation: XmNcommandWindow resource in XmMainWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MainW.c

EZYXN23W Negative margin value ignored.

Explanation: XmMainWindow widget class.

System Action: The application continues.

User or Operator Response: None.

EZYXN31W • EZYXN34W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MainW.c

EZYXN31W MenuShell widgets must have a xmRowColumnWidgetClass child.

Explanation: XmMenuShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN32W Attempting to manage an incomplete menu.

Explanation: XmMenuShell widget class. XmCreatePopupMenu function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN33W XmPopup requires a subclass of shellWidgetClass.

Explanation: The _XmPopupI function detected an error in XtIsShell. The passed Widget was of an incorrect class.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN34W XmPopdown requires a subclass of shellWidgetClass.

Explanation: The _XmPopdown function detected an error in XtIsShell. The passed Widget was of an incorrect class.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN35W XtMenuPopup requires exactly one argument.

Explanation: The _XmMenuPopupAction function detected that the argument count was not one.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN36W XtMenuPopup only supports ButtonPress, KeyPress or EnterNotify events.

Explanation: The _XmMenuPopupAction function detected that the XEvent did not match the allowed values of ButtonPress, KeyPress or EnterNotify. The modal grab will be Nonexclusive.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN37W Cannot find popup widget *string* in XtMenuPopup.

Explanation: The _XmMenuPopupAction function detected a failure in the _XmFindPopup routine. The specified popup was not found.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN38W Cannot find popup widget *string* in XtMenuPopdown.

Explanation: The _XmMenuPopdownAction function detected a failure in the _XmFindPopup routine. The specified popup was not found.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN39W XtMenuPopdown called with more than one argument.

Explanation: The _XmMenuPopdownAction function detected that the number of arguments was more than one.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

EZYXN40W • EZYXN43W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN40W Cannot change XmNlayoutDirection after initialization.

Explanation: The SetValues function detected that the XmNlayoutDirection value was changed after initialization. The XmNlayoutDirection value is reset to the initialized value.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: MenuShell.c

EZYXN41W Invalid Dialog Type.

Explanation: XmNdialogType resource in XmMessageBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN42W Invalid Default Button Type.

Explanation: XmNdefaultButtonType resource in XmMessageBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN43W Invalid Alignment Type.

Explanation: XmNmessageAlignment resource in XmMessageBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN44W Invalid Child Type.

Explanation: XmMessageBoxGetChild function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN45W PushButton Id cannot be changed directly.

Explanation: XmMessageBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN46W Use XmNdefaultButtonType to set MessageBox default button.

Explanation: XmMessageBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: MessageB.c

EZYXN54W Invalid minimum value, must be greater than zero.

Explanation: XmNpaneMinimum resource in XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN55W Invalid maximum value, must be greater than zero.

Explanation: XmNpaneMaximum resource in XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN56W Invalid minimum/maximum value, minimum value must be smaller than the maximum value.

Explanation: XmNpaneMinimum resource in XmPanedWindow widget class. XmNpaneMaximum resource in XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN57W Constraints do not allow appropriate sizing.

Explanation: XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN58W Too few parameters.

Explanation: XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN59W Invalid 1st parameter.

Explanation: XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN61W fontList is not defined.

Explanation: XmPanedWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PanedW.c

EZYXN71W Must be a vendor shell.

Explanation: XmAddProtocols function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Protocols.c

EZYXN72W Protocol manager already exists.

Explanation: XmAddProtocols function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Protocols.c

EZYXN73W More protocols than I can handle.

Explanation: XmAddProtocols function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Protocols.c

EZYXN81W Not enough memory.

Explanation: There was not enough memory to perform the function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PushB.c

EZYXO01W Attempt to set width to zero. Set to default value 16.

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO02W Attempt to set width to zero. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO03W Attempt to set height to zero. Set to default value 16.

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO04W Attempt to set height to zero. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO05W XmNhlpWidget not used by PopUps. It is set to NULL.

Explanation: XmNmnuHelpWidget resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO06W XmNhelpWidget not used by Pulldowns. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO07W XmNhelpWidget not used by Option menus. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO08W XmNhelpWidget not used by Work Areas. It is set to NULL.

Explanation: XmNmenuHelpWidget resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO09W Unknown value of XmNrowColumnType. It is set to WorkArea.

Explanation: XmNrowColumnType resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO10W Widget hierarchy not appropriate for this XmNrowColumnType. It is set to WorkArea.

Explanation: XmCreatePulldownMenu function.

System Action: The application continues.

User or Operator Response: None.

EZYXO11W • EZYXO14W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO11W Attempt to change XmNrowColumnType after initialization. The value is ignored.

Explanation: XmNrowColumnType resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO12W Unknown value of XmNorientation. The default value is used.

Explanation: XmNorientation resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO13W Attempt to set XmNorientation to unknown value. The value is ignored.

Explanation: XmNorientation resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO14W Unknown value of XmNpacking. The default value is used.

Explanation: XmNpacking resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO15W Attempt to set XmNpacking to unknown value. The value is ignored.

Explanation: XmNpacking resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO16W Unknown value of XmNentryAlignment. The default value is used.

Explanation: XmNentryAlignment resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO17W Attempt to set XmNentryAlignment to unknown value. The value is ignored.

Explanation: XmNentryAlignment resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO18W Attempt to set XmNisHomogenous to FALSE for a RowColumn widget of type XmMENU_BAR. The value is ignored.

Explanation: XmNisHomogeneous resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO19W Attempt to change XmNentryClass for a RowColumn widget of type XmMENU_BAR. The value is ignored.

Explanation: XmNentryClass resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO20W Attempt to change XmNwhichButton via XtSetValues for a RowColumn widget of type XmMENU_PULLDOWN. The value is ignored.

Explanation: XmNwhichButton resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO21W Attempt to change XmNmenuPost via XtSetValues for a RowColumn widget of type XmMENU_PULLDOWN. The value is ignored.

Explanation: XmNmenuPost resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO22W Attempt to set XmNpostMenu to an illegal value. The value is ignored.

Explanation: XmNmenuPost resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO23W Attempt to change XmNshadowThickness for a RowColumn widget not of type XmMENU_PULLDOWN or XmMENU_POPUP. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO24W Attempt to change XmNorientation for a RowColumn widget of type XmMENU_OPTION. The value is ignored.

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO25W Attempt to add wrong type child to a menu (i.e. RowColumn) widget

Explanation: XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO26W Attempt to add wrong type child to a homogeneous RowColumn widget.

Explanation: XmNisHomogeneous resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO27W Attempt to change XmNisHomogeneous for a RowColumn widget of type XmMENU_OPTION ignored.

Explanation: XmNisHomogeneous resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO28W Tear off enabled on a shared menupane is allowed but not recommended.

Explanation: Tear-off Menus in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

EZYXO29W • EZYXO33W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO29W Illegal mnemonic character. Could not convert X KEYSYM to a keycode.

Explanation: XmNmnemonic resource in XmRowColumn widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RowColumn.c

EZYXO31W The scale minimum value is greater than or equal to the scale maximum value.

Explanation: XmNminimum resource in XmScale widget class. XmNmaximum resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO32W The specified scale value is less than the minimum scale value.

Explanation: XmNvalue resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO33W The specified scale value is greater than the maximum scale value.

Explanation: XmNvalue resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO34W Incorrect orientation.

Explanation: XmNorientation resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO35W Incorrect processing direction.

Explanation: XmNprocessingDirection resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO36W Invalid highlight thickness.

Explanation: XmNhighlightThickness resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO37W Invalid scaleMultiple; greater than (max - min).

Explanation: XmNscaleMultiple resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO38W Invalid scaleMultiple; less than zero.

Explanation: XmNscaleMultiple resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO39W (Maximum - minimum) cannot be greater than INT_MAX divided by 2. Minimum has been set to zero. Maximum may have been set to (INT_MAX/2).

Explanation: XmNmaximum resource in XmScale widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Scale.c

EZYXO41W The scrollbar minimum value is greater than or equal to the scrollbar maximum value.

Explanation: XmNminimum resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO42W The specified slider size is less than 1.

Explanation: XmNincrement resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO43W The specified scrollbar value is less than the minimum scrollbar value.

Explanation: XmNvalue resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO44W The specified scrollbar value is greater than the maximum scrollbar value minus the scrollbar slider size.

Explanation: XmNvalue resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO45W Incorrect orientation.

Explanation: XmNorientation resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO46W Incorrect processing direction.

Explanation: XmNprocessingDirection resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO47W The scrollbar increment is less than 1.

Explanation: XmNincrement resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO48W The scrollbar page increment is less than 1.

Explanation: XmNpageIncrement resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO49W The scrollbar initial delay is less than 1.

Explanation: XmNinitialDelay resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO50W The scrollbar repeat delay is less than 1.

Explanation: XmNrepeatDelay resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO51W Error in context manager; scrollbar backgrounds cannot be set correctly

Explanation: XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO52W Error in context manager; scrollbar foregrounds cannot be set correctly.

Explanation: XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO53W Specified slider size is greater than the scrollbar maximum value minus the scrollbar minimum value.

Explanation: XmNsliderSize resource in XmScrollBar widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollBar.c

EZYXO61W Invalid ScrollBar Display policy.

Explanation: XmNscrollBarDisplayPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO62W Invalid Scrolling Policy.

Explanation: XmNscrollingPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO63W Invalid Visual Policy.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO64W Invalid placement policy.

Explanation: XmNscrollBarPlacement resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO66W Cannot change scrolling policy after initialization.

Explanation: XmNscrollingPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

EZYXO67W • EZYXO70W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO67W Cannot change visual policy after initialization.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO68W Cannot set AS_NEEDED scrollbar policy with a visual policy of VARIABLE.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO69W Cannot change scrollbar widget in AUTOMATIC mode.

Explanation: XmNscrollingPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYXO70W Cannot change clip window.

Explanation: XmNclipWindow resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYX071W Cannot set visual policy of CONSTANT in APPLICATION_DEFINED mode.

Explanation: XmNvisualPolicy resource in XmScrolledWindow widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYX072W Wrong parameters passed to the function.

Explanation: XmScrollVisible function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrolledW.c

EZYX081W Incorrect dialog type.

Explanation: XmNdialogType resource in XmSelectionBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYX082W Dialog type cannot be modified.

Explanation: XmNdialogType resource in XmSelectionBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYX083W Only one work area child allowed.

Explanation: XmSelectionBox widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

EZYXO84W • EZYXP02W

Procedure Name: SelectioB.c

EZYXO84W Invalid child type.

Explanation: XmSelectionBoxGetChild function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SelectioB.c

EZYXO91W Invalid separator type.

Explanation: XmNseparatorType resource in XmSeparator widget class. XmNseparatorType resource in XmSeparatorGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Separator.c

EZYXO92W Invalid orientation.

Explanation: XmNorientation resource in XmSeparator widget class. XmNorientation resource in XmSeparatorGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Separator.c

EZYXP01W Invalid source, source ignored.

Explanation: XmNsource resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Text.c

EZYXP02W Invalid edit mode.

Explanation: XmNeditMode resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Text.c

EZYXP03W Text widget is editable; traversalOn must be true.

Explanation: XmNeditable resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Text.c

EZYXP04W Can't find position in MovePreviousLine().

Explanation: XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextIn.c

EZYXP05W Invalid rows, must be > 0.

Explanation: XmNrows resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextOut.c

EZYXP06W XmFontListInitFontContext failed.

Explanation: XmNfontList resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextFc, TextOut.c

EZYXP07W XmFontListGetNextFont failed.

Explanation: XmNfontList resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c, TextOut.c

EZYXP08W Character *char* is not supported in font. It is discarded.

Explanation: XmNfontList resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP09W String *string* is not supported in font. It is discarded.

Explanation: XmNfontList resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP10W Cannot use a multibyte locale without a fontset. The value is discarded.

Explanation: XmNfontList resource in XmText widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP11W Invalid cursor position, must be ≥ 0 .

Explanation: XmNcursorPosition resource in XmTextField widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP12W Invalid columns, must be > 0.

Explanation: XmNcolumns resource in XmText widget class. XmNcolumns resource in XmTextField widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c, TextOut.c

EZYXP13W XmNtraversalOn must always be true.

Explanation: XmTextField widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP14W Invalid columns, must be >= 0.

Explanation: XmNcolumns resource in XmTextField widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: TextF.c

EZYXP21W Indicator type should be either XmONE_OF_MANY or XmN_OF_MANY.

Explanation: XmNindicatorType resource in XmToggleButton widget class. XmNindicatorType resource in XmToggleButtonGadget widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ToggleB.c

EZYXP31W Invalid value for navigation_type.

Explanation: XmNnavigationType resource in XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

EZYXP32W • EZYXP35W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP32W Wrong value in old for navigation_type!!

Explanation: XmNnavigationType resource in XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP33W Traversal bootstrap situation with bad parameters.

Explanation: XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP34W Attempt to traverse to new tab using bad parameters.

Explanation: XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP35W startWidget is not in child list.

Explanation: XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP36W Bad parameters to TraverseToChild.

Explanation: XmManager widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Traversal.c

EZYXP41W Invalid value for delete response.

Explanation: XmNdeleteResponse resource in VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP42W Invalid XmNpreeditType, default to OverTheSpot.

Explanation: XmNpreeditType resource in VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP43W Invalid value for XmNinputPolicy.

Explanation: The VendorExtSetValues function detected an incorrect value for the XmVendorShellExtPartPtr input policy. The policy is required to be either XmPER_SHELL or XmPER_WIDGET.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP44W XmNlayoutDirection cannot be changed.

Explanation: The VendorExtSetValues function detected that the XmNlayoutDirection value was changed after initialization. The XmNlayoutDirection value is reset to the initialized value.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

EZYXP45W • EZYXP48E

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP45W FetchUnitType: bad widget class.

Explanation: VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorSE.c

EZYXP46W String to no-op conversion needs no extra arguments.

Explanation: VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorSE.c

EZYXP47W FetchUnitType called without a widget to reference.

Explanation: VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorSE.c

EZYXP48E Fatal Error: _XmGetDefaultDisplay cannot be used prior to VendorS.Initialize, returns NULL.

Explanation: _XmDisplayHandle was not set before the _XmGetDefaultDisplay function was called. A NULL value is returned.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXP51W Virtual bindings Initialize hasn't been called.

Explanation: XmVirtKeys widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VirtKeys.c

EZYXP61W Invalid color requested from _XmAccessColorData.

Explanation: XmGetColors function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Visual.c

EZYXP62W Cannot allocate colormap entry for background, setting background to white.

Explanation: XmNbackground resource in Core widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Visual.c

EZYXP63W Cannot parse given background color, setting background to white.

Explanation: XmNbackground resource in Core widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Visual.c

EZYXP71W The specified Input Method failed to init : string.

Explanation: XmNinputMethod resource in VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP72W Cannot create the Input Method Object.

Explanation: XmNinputMethod resource in VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP73W XmIMFocus invoked with NULL widget.

Explanation: XmNinputMethod resource in VendorShell widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP74W XmIMMove invoked without the Input Method focus.

Explanation: XmImGetXIM function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmIm.c

EZYXP81W No context found for extension.

Explanation: When trying to map the extension to a context, no context was found.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP82W _XmPopWidgetExtData: no extension found with XFindContext.

Explanation: This is a debug message.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP83W **XmFreeWidgetExtData is an unsupported routine.**

Explanation: This routine is no longer a supported interface.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP84W **getLabelSecResData: not enough memory.**

Explanation: There was not enough memory to perform the requested function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: BaseClass.c

EZYXP91W **Creating multiple XmDisplays for the same X display. Only the first XmDisplay created for a particular X display can be referenced by calls to XmGetXmDisplay.**

Explanation: XmDisplay widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Display.c

EZYXP92W **Received TOP_LEVEL_LEAVE with no active DragContext.**

Explanation: Function requested TOP_LEVEL_LEAVE but there was no DragContext found.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Display.c

EZYXP93W Cannot set XmDisplay class to a non-subclass of XmDisplay.

Explanation: XmDisplayClass widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Display.c

EZYXQ01W The _MOTIF_DRAG_WINDOW has been destroyed.

Explanation: The window is no longer a valid context.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ02W The protocol version levels do not match.

Explanation: The level of the Motif server and client code drag protocol is not the same.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c, DragICC.c

EZYXQ03W Unable to open display.

Explanation: XmInternAtom function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ04W The atoms table is empty.

Explanation: WriteAtomsTable function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ05W The target table is empty.

Explanation: WriteTargetsTable function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ06W The target table has an inconsistent property.

Explanation: ReadTargetsTable function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ07W Invalid target table index.

Explanation: ReadTargetsTable function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragBS.c

EZYXQ11W GenerateCallback does not expect XmCR_DROP_SITE_ENTR as a reason.

Explanation: GenerateClientCallback function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ12W Invalid selection in DropConvertCallback.

Explanation: DropConvertCallback function.

System Action: The application continues.

User or Operator Response: None.

EZYXQ13W • EZYXQ16W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ13W The drop selection was lost.

Explanation: DropLoseSelection function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ14W XGrabPointer failed.

Explanation: Call to lower level function, XGrabPointer, was unsuccessful.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ15W ExternalNotifyHandler: the callback reason is not acceptable.

Explanation: XmDragStart function. The drag message did not contain a defined reason code.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ16W XmDragStart must be called as a result of a button press.

Explanation: XmDragStart function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragC.c

EZYXQ21W Unknown drag and drop message type.

Explanation: Message type received was not expected in this context.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragICC.c

EZYXQ22W The protocol version levels do not match.

Explanation: The `_XmGetDragReceiverInfo` function detected that the `xmDragReceiverInfoStruct` protocol version did not match the `_MOTIF_DRAG_PROTOCOL_VERSION`.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: DragICC.c

EZYXQ31W No geometry specified for dragIcon pixmap.

Explanation: `XmDragIcon` widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragIcon.c

EZYXQ32W dragIcon created with no pixmap.

Explanation: `XmNpixmap` resource in `XmDragIcon` widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragIcon.c

EZYXQ33W String to Bitmap converter needs Screen argument.

Explanation: `XmCvtStringToBitmap` function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

EZYXQ41W • EZYXQ51W

Source Data Set: Xm.a library.

Procedure Name: DragIcon.c

EZYXQ41W Depth mismatch.

Explanation: XmNblendModel resource in XmDragContext widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ42W Unknown icon attachment.

Explanation: XmNattachment resource in XmDragIcon widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ43W Unknown drag state.

Explanation: GetDragIconColors function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ44W Unknown blendModel.

Explanation: XmNblendModel resource in XmDragContext widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragOverS.c

EZYXQ51W Unable to get dropSite window geometry.

Explanation: CreateAnimationSaveData function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragUnder.c

EZYXQ52W Invalid animationPixmapDepth.

Explanation: XmNanimationPixmapDepth resource in XmDropSite registry.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DragUnder.c

EZYXQ61W Cannot create drop sites which are children of a simple drop site.

Explanation: XmNdropsiteType resource in XmDropSite registry.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ62W Receiving Motion Events without an active drag context.

Explanation: XmDragContext widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ63W Receiving operation changed without an active drag context.

Explanation: XmDragContext widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ64W Creating an active drop site with no drop procedure.

Explanation: XmDropSiteRegister function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ65W Cannot set rectangles or rectangle numbers of composite drop sites.

Explanation: XmNdropsRectangles resource in XmDropSite registry.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ66W Registering a widget as a drop site out of sequence. Ancestors must be registered before any of their descendants are registered.

Explanation: XmDropSiteRegister function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ67W Cannot register widget as a drop site more than once.

Explanation: XmDropSiteRegister function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ68W DropSite type may only be set at creation time.

Explanation: XmNdropsSiteType resource in XmDropSite registry.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ69W Cannot change rectangles of non-simple dropsite.

Explanation: XmNdropRectangles resource in XmDropSite registry.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgr.c

EZYXQ70W Cannot register a drop site which is a descendent of a simple drop site.

Explanation: XmDropSiteRegister function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrL.c

EZYXQ71W Cannot create a discontiguous child list for a composite drop site.

Explanation: XmDropSiteRegister function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrL.c

EZYXQ72W *string* is not a drop site child of *string*.

Explanation: XmDropSiteRegister function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrL.c

EZYXQ73W Cannot register a Shell as a drop site.

Explanation: The XmDropSiteRegister function detected an error while attempting to register a drop site. The Widget to be registered is defined as a Shell.

System Action: The application continues.

EZYXQ81W • EZYXR01W

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: DropSMgrI.c

EZYXQ81W Failure of geometry request to "almost" reply.

Explanation: The Intrinsics protocol guarantees a "Yes" response to a request with identical geometry to that which was returned by a previous request returning "almost".

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: GeoUtils.c

EZYXQ82W Invalid order found in XmSelectionBox.

Explanation: XmSelectionBox geometry.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: GeoUtils.c

EZYXQ91W Memory error.

Explanation: An attempt to allocate memory failed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Increase the region size in which the program executes.

Source Data Set: Xm.a library.

Procedure Name: Region.c

EZYXR01W Illegal representation type id.

Explanation: XmRepTypeValidValue function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RepType.c

EZYXR02W Illegal value *number* for rep type XmRxxx.

Explanation: XmRepTypeValidValue function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: RepType.c

EZYXR03W Reverse Conversion of

Explanation: This is a header message for XmRepTypeEntry conversions in the ReverseConvertRepType function, and will be output for most conversions.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: RepType.c

EZYXR11W FetchUnitType: bad widget class.

Explanation: FetchUnitType function. Widget is not gadget, manager, or primitive.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR12W Cannot continue because of errors in a default font list.

Explanation: GetNextFontListEntry function failed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR13W Missing colon in font string *string*; any remaining fonts in list unparsed.

Explanation: The GetNextFontListEntry function detected that a font set in the font list was missing the colon delimiter. Any fonts remaining on the list are not parsed.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

EZYXR14W • EZYXR17W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR14W Invalid delimiter in tag *string*; any remaining fonts in list unparsed.

Explanation: The GetNextFontListEntry function detected that the delimiter between two font list items is not valid. Any fonts remaining in the list are not parsed.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR15W Unmatched quotation marks in tag *string*; any remaing fonts in list unparsed.

Explanation: The GetFontName function detected that the font list started with a quotation mark ("), but did not end with a quotation mark ("). Remaining fonts in the list are not parsed.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR16W Null tag found when converting to type *string*; any remaining fonts in list unparsed.

Explanation: The GetFontTag function found a NULL tag when expecting a real value. Remaining fonts in the list are not parsed.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR17W Cannot convert XmString to Compound Text.

Explanation: The XmCvtXmStringToCT function detected an error in the cvtXmStringToText function. The XmString could not be converted directly to a Compound String.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResEncod.c

EZYXR18W Insufficient memory for XmbTextListToTextProperty.

Explanation: The processCharsetAndText function detected that the XNoMemory flag was turned on during XmbTextListToTextProperty processing.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResEncod.c

EZYXR19W Locale not supported for XmbTextListToTextProperty.

Explanation: The processCharsetAndText function detected that the XLocaleNotSupported flag was turned on during XmbTextListToTextProperty processing.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResEncod.c

EZYXR20W XmbTextListToTextProperty failed.

Explanation: The processCharsetAndText function detected a failure in the XmbTextListToTextProperty function.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResEncod.c

EZYXR21W Icon screen mismatch.

Explanation: XmScreen widget class.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Screen.c

EZYXR22W Cannot get XmScreen because XmDisplay was not found.

Explanation: XmGetXmScreen function.

System Action: The application continues.

User or Operator Response: None.

EZYXR41W • EZYXR52W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Screen.c

EZYXR41W Could not allocate memory for color object data.

Explanation: An attempt to allocate memory failed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Increase the region size in which the program executes.

Source Data Set: Xm.a library.

Procedure Name: ColorObj.c

EZYXR42W Bad screen number from color server selection.

Explanation: GetSelection function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ColorObj.c CDE Icon Gadget messages

EZYXR51W Cannot convert widget name to Widget.

Explanation: The StringToEntity function detected that an incorrect number of parameters was used.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResConvert.c

EZYXR52W Cannot convert compound text to XmString.

Explanation: The XmCvtTextToXmString function detected an error while converting a compound text string to an XmString.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResEncod.c

EZYXR53W Cannot convert XmString to compound text.

Explanation: The XmCvtXmStringToText function detected an error while converting the XmString to an ASCII string.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ResEncod.c

EZYXR61W Applications cannot add children to XmComboBox widgets.

Explanation: The InsertChild function detected that the application attempted to add another child after the ComboBox had already made its children.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR62W XmNcomboBoxType resource cannot be changed by XtSetValues.

Explanation: The SetValues function detected that the ComboBox type was illegally changed after creation. The ComboBox type is reset to the original value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR63W Internal widget has been destroyed. Behavior is undefined.

Explanation: The ListSelectionCB function or the DoLayout function detected that the Widget passed into the function is missing or not valid.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR64W Internal widget has been unmanaged. Behavior is undefined.

Explanation: The DoLayout function detected that either the List or EditText XmComboBoxWidget is unmanaged.

System Action: The application continues.

User or Operator Response: None.

EZYXR65W • EZYXR67W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR65W XmQUICK_NAVIGATE is only valid for ComboBoxes of XmNcomboBoxType XmDROP_DOWN_LIST.

Explanation: The Initialize function or the SetValues function detected that XmQUICK_NAVIGATE was not valid with XmNcomboBoxType XmDROP_DOWN_LIST. The XmNmatchBehavior is reset to its previous state.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR66W Action invoked with the wrong number of parameters.

Explanation: The CBListAction function detected that an incorrect number of parameters was used. The required number of parameters is one.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR67W Action routine called from a widget that is not a descendant of ComboBox.

Explanation: The FindComboBox call failed in one of the following functions because the passed Widget did not contain a ComboBox:

- CBArmAndDropDownList
- CBDisarm
- CBDropDownList
- CBFocusIn
- CBFocusOut
- CBTextFocusOut
- CBActivate, CBCancel
- CBListAction

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error. error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR68W XmComboBoxSelectItem called with an item not present in the ComboBox.

Explanation: The XmComboBoxSelectItem function detected that the selected item is not present in the ComboBox.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR69W XmComboBoxSetItem called with an item present in the ComboBox.

Explanation: The XmComboBoxSetItem function detected a failure in the XmListItemPos function due to an incorrect XmString value being passed into either XmComboBoxSetItem or XmListItemPos.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR70W XmComboBoxDeletePos called with an invalid position.

Explanation: The XmComboBoxDeletePos function detected one of the following:

- The position to be deleted had a value lower than zero.
- The position to be deleted was greater than the number of items in the list.
- The number of list items was zero.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR71W XmComboBox utility routine called with an invalid widget.

Explanation: One of the following functions detected that passed Widget is not a ComboBox:

- XmComboBoxAddItem
- XmComboBoxDeletePos
- XmComboBoxSelectItem
- XmComboBoxSetItem
- XmComboBoxUpdate

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR72W Applications may not set the automatic XmComboBox widget children.

Explanation: The Initialize function or the SetValues function detected that a change in either the list child or the editbox child occurred. The children are reset to NULL.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ComboBox.c

EZYXR81W Action invoked with the wrong number of parameters.

Explanation: One of the following functions was invoked with the wrong number of parameters:

- ContainerHandleBtn1Down
- ContainerHandleBtn1Motion
- ContainerHandleBtn1Up
- ContainerHandleBtn2Down
- ContainerHandleBtn2Motion

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Container.c

EZYXR82W XmNdetailColumnHeading and XmNdetailColumnHeadingCount do not match!

Explanation: The Initialize function detected that the XmNdetailColumnHeading and XmNdetailColumnHeadingCount do not match.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Container.c

EZYXR91W Widget class *string* has invalid CompositeClassExtension record.

Explanation: The FindCompClassExtension function detected that the current Widget has incorrect version and record_size values in the CompositeClassExtension record.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Manager.c

EZYXR92W Cannot change XmNlayoutDirection or XmNstringDirection after initialization.

Explanation: The SetValues function detected that either the XmNlayoutDirection or XmNstringDirection was changed after initialization. The incorrect value is reset to its initialized value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Manager.c

EZYXS01W XmNnotebookChildType resource cannot be set by XtSetValues.

Explanation: The ConstraintSetValues function detected that the XmNnotebookChildType does not match the initialized value. The XmNnotebookChildType is reset to its original value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Notebook.c

EZYXS11W Wrong number of parameters for CvtStringToIconPixmap.

Explanation: The CvtStringToIconPixmap function detected that an incorrect number of parameters was used.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: PixConv.c

EZYXS21W Cannot change XmNlayoutDirection after initialization.

Explanation: The SetValues function detected that the XmNlayoutDirection value was changed after initialization. The XmNlayoutDirection value is reset to its initialized value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Primitive.c

EZYXS31W AssocNavigator requires a navigator trait.

Explanation: The _XmSFAddNavigator function detected that the XmNavigatorTrait value was set to NULL.

System Action: The application continues.

User or Operator Response: None.

EZYXS32W • EZYXS43W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollFrameT.c

EZYXS32W DeAssocNavigator requires a navigator trait.

Explanation: The _XmSFRemoveNavigator function detected that the XmNavigatorTrait value was set to NULL.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: ScrollFrameT.c

EZYXS41W No items supplied for XmSTRING child.

Explanation: The ConstraintInitialize function or the ConstraintSetValues function detected that there were no items placed in the XmStringTable, but there were items passed to the function.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SpinB.c

EZYXS42W XmNincrementValue cannot be 0. A value of 1 will be used.

Explanation: The ConstraintInitialize function or the ConstraintSetValues function detected an incorrect XmNincrementValue. XmNincrementValue is reset to one.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SpinB.c

EZYXS43W Spin direction specified by XmNincrementValue has been reversed to match the specified XmNminimumValue and XmNmaximumValue.

Explanation: The ConstraintInitialize function or the ConstraintSetValues function detected that the Spin direction was incorrect. The spin direction was reversed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SpinB.c

EZYXS44W XmNposition out of range. Minimum XmNposition used.

Explanation: The ValidatePositionValue function detected that XmNposition was lower than the minimum allowed value. XmNposition is reset to the minimum value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SpinB.c

EZYXS45W XmNposition out of range. Maximum XmNposition used.

Explanation: The ValidatePositionValue function detected that XmNposition was greater than the maximum allowed value. XmNposition is reset to the maximum value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SpinB.c

EZYXS46W Invalid value for XmNpositionType. Using default value.

Explanation: The ConstraintInitialize or the ConstraintSetValues function detected that the XmNpositionType value is incorrect.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SpinB.c

EZYXS51W Calling SelectionCallbackWrapper when transfers should be finished

Explanation: The SelectionCallbackWrapper function was called when all outstanding transfers had completed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS52W The format and type of the callback supplied data does not match the data being merged.

Explanation: The XmConvertMerge function detected that the format and type of the callback supplied data does not match the format and type of the data to be merged.

System Action: The application continues.

User or Operator Response: None.

EZYXS53W • EZYXS56W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS53W The status in the `XmConvertCallbackStruct` is not `XmCONVERT_MERGE`.

Explanation: The `XmConvertMerge` function detected that the `XmCONVERT_MERGE` was not set, but it is required.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS54W `XmCONVERT_MORE` is not supported.

Explanation: The `_XmConvertHandler` function detected that `XmCONVERT_MORE` is being used, which is not supported by this release of the Motif specification. The flag is changed to `XmCONVERT_DEFAULT`.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS55W Bad atom value found.

Explanation: The `GetSafeAtomName` function detected that `XGetAtomName` returned with either `NULL` or an incorrect atom name.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS56W Warning: Attempt to start a `MULTIPLE` transfer when one is in progress.

Explanation: The `XmTransferStartRequest` function detected a `MULTIPLE` request while processing the current transfer.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS57W Warning: Attempt to send a MULTIPLE transfer when one is not in progress.

Explanation: The XmTransferSendRequest function detected that the attempted MULTIPLE request is not allowed for this transfer.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: Transfer.c

EZYXS61W XtVaTypedArg conversion needs non-null widget handle.

Explanation: The Widget passed to the _XmTypedArgToArg function was NULL.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VaSimple.c

EZYXS62W Unable to find type of resource for conversion.

Explanation: The _XmTypedArgToArg function detected that the passed resource type in the typed argument list did not match a known resource type.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VaSimple.c

EZYXS63W Type conversion failed.

Explanation: The _XmTypedArgToArg function detected that the address of the receiving XrmValue variable was NULL. The XtConvert process failed.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VaSimple.c

EZYXS71W Name: *string* Class: *string*

Explanation: Motif Generic Header message called within VendorS.c for use with applications from the previous release of Motif.

System Action: The application continues.

User or Operator Response: None.

EZYXS72W • EZYXS83W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: VendorS.c

EZYXS72W Action invoked with the wrong number of parameters.

Explanation: A function was called with one of the following:

- No arguments
- More than one argument
- An argument that was not valid

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: FileSB.c, Notebook.c, ScrollBar.c, SelectioB.c

EZYXS81W XmNtag cannot be NULL. Setting to empty string.

Explanation: The ValidateTag function detected that the XmNtag was set to NULL.

System Action: XmNtag is set to default and the application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmRenderT.c

EZYXS82W Display is NULL. Cannot load font.

Explanation: The ValidateAndLoadFont function detected that there was no valid Display to which to load the font.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmRenderT.c

EZYXS83W XmNfontType invalid. Cannot load font.

Explanation: The ValidateAndLoadFont function detected that the specified font or fontset is not a valid type.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmRenderT.c

EZYXS84W Conversion failed. Cannot load font.

Explanation: The ValidateAndLoadFont function detected that the font conversion failed because no callback was issued. The font did not load.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmRenderT.c

EZYXS85W XmNfontType set to XmAS_IS. Cannot load font.

Explanation: The ValidateAndLoadFont function detected that the fontType was set to XmAS_IS. The specified font is not loaded.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmRenderT.c

EZYXS86W XmNloadModel is XmLOAD_IMMEDIATE but XmNfont and XmNfontName not specified. Cannot load font.

Explanation: The ValidateAndLoadFont function detected that the font and font name were not specified. The font was not loaded.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmRenderT.c

EZYXS91W No font found.

Explanation: The OptLineMetrics function or the SpecifiedSegmentExtents function detected that a font was not specified.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmString.c

EZYXT01W Tab value cannot be negative.

Explanation: The XmTabCreate function or the XmTabSetValue function detected that the _XmTabValue contained a negative value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: XmTabList.c

EZYXT11W XmNtextField resource cannot be set.

Explanation: The SetValues function detected that the current textfield Widget is not equal to the new textfield Widget, and cannot be set to the specified value.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SSpinB.c

EZYXT12W XmNpositionType resource can only be set at creation time.

Explanation: The SetValues function detected that the current position of XmPOSITION_{ARRAY,VALUE} does not equal the new position. The position cannot be changed.

System Action: The application continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SSpinB.c

EZYXT13W Item does not exist. XmNposition is unchanged.

Explanation: The XmSimpleSpinBoxSetItem function attempted to access an XmNposition value that is no longer present.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xm.a library.

Procedure Name: SSpinB.c

EZYXU01W The duplicate option *string* was ignored

Explanation: The same command line option has been repeated more than once.

System Action: The compiler continues.

User or Operator Response: Remove duplicate command line option.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU02W The unknown option *string* was ignored

Explanation: An unknown option has been used in the compiler command line.

System Action: The compiler continues.

User or Operator Response: Check what you typed on the command line.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU03E The additional UIL source file: *string* was ignored

Explanation: More than one source file was specified. Only the first source file will be compiled.

System Action: The compiler continues.

User or Operator Response: Compile additional source files using separate invocations or the compiler.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU04S An error occurred opening source file: *string*.

Explanation: The source file specified could not be opened.

System Action: The compiler ends.

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c, UilSrcSrc.c

Procedure Name: various

EZYXU05S An error occurred reading next line of source file: *string*.

Explanation: The source file specified could not be successfully read.

System Action: The compiler ends.

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c, UilSrcSrc.c

Procedure Name: various

EZYXU06S An internal error occurred in: *string*

Explanation: The compiler detected an internal error.

System Action: The compiler ends.

User or Operator Response: Submit a software problem report.

System Programmer Response: None.

EZYXU07E • EZYXU11E

Source Data Set: UilDiag.c

Procedure Name: diag_handler

EZYXU07E The line was truncated at *number* characters

Explanation: The compiler encountered a source line greater than 132 characters. Characters beyond the 132 character limit were ignored.

System Action: The compiler continues.

User or Operator Response: Break each source line longer than 132 characters into several source lines. Long string literals can be created using the concatenation operator.

System Programmer Response: None.

Source Data Set: UilSrcSrc.c

Procedure Name: src_get_source_line

EZYXU08E The value of *string* is out of range *string*.

Explanation: The value specified is outside the legal range for its type.

System Action: The compiler continues.

User or Operator Response: Change the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c, UilSarMod.c, UilSemVal.c

Procedure Name: various

EZYXU09E *string1* not terminated *string2*

Explanation: A string was not properly terminated. String1 and string2 explain the defect.

System Action: The compiler continues.

User or Operator Response: Change the UIL module source to terminate the sequence properly.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: various

EZYXU10E The unprintable character *char* ignored

Explanation: The compiler encountered an illegal control character in the UIL specification file. The decimal value of the character is displayed.

System Action: The compiler continues.

User or Operator Response: Replace the character with the sequence specified in the message.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU11E The unknown sequence *string* ignored

Explanation: The compiler detected a sequence of printable characters it did not understand. The compiler omitted the sequence of characters shown.

System Action: The compiler continues.

User or Operator Response: Fix the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU12E The unknown escape sequence `\char - \\` ignored

Explanation: A back slash was followed by an unknown escape character. The back slash is the escape character in UIL. A selected set of single characters can follow a back slash such as `\n` for new line. The character following the back slash was not one of the selected set.

System Action: The compiler continues.

User or Operator Response: Fix the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU13E The name exceeds *number* characters. It was truncated to: *string*

Explanation: The UIL compiler encountered a name longer than 31 characters. The compiler truncated the name to the leftmost 31 characters.

System Action: The compiler continues.

User or Operator Response: Fix the UIL module source.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yylex

EZYXU14S The compiler ran out of virtual memory

Explanation: The compiler ran out of virtual memory.

System Action: The compiler continues.

User or Operator Response: Reduce the size of your application or increase virtual memory.

System Programmer Response: None.

Source Data Set: UilMain.c

Procedure Name: uil_mmove

EZYXU15E An unexpected string token was seen. Parsing will resume after *string*.

Explanation: At the point marked in the module, the compiler found a construct, such as a punctuation mark, name, or keyword, when it was expecting a different construct. The compiler continued analyzing the module at the next occurrence of the construct stated in the message.

System Action: The compiler continues.

User or Operator Response: Check the syntax of your UIL module at the point marked by the compiler. If the module specifies case-sensitive names, check that your keywords are in lowercase characters.

System Programmer Response: None.

Source Data Set: UilLexAna.c, UilLexPars.c

Procedure Name: various

EZYXU16E *string string must be defined before this reference*

Explanation: The widget pointed to in the message was either never defined or not defined prior to this point in the module. The compiler requires to be defined before you refer to the widget at the next occurrence of the construct stated in the message.

System Action: The compiler continues.

User or Operator Response: Check for a misspelling of the name of the widget, a missing declaration for the widget, or declaring the widget after its first reference. If names in the module are case-sensitive, the spellings of the name in the declaration and in the reference must match exactly.

System Programmer Response: None.

Source Data Set: UilSarProc.c, UilSarVal.c

Procedure Name: various

EZYXU17E *The context requires a string but string was specified*

Explanation: At the point marked in the specification, one type of object (such as a widget) is required and your specification supplied a different type of object (such as a value).

System Action: The compiler continues.

User or Operator Response: Check for misspelling. Also check to ensure that you have referred to the intended widget.

System Programmer Response: None.

Source Data Set: UilSarProc.c, UilSarVal.c

Procedure Name: various

EZYXU18E *string is not implemented yet*

Explanation: You are using a feature of UIL that has not been implemented yet.

System Action: The compiler continues.

User or Operator Response: Try an alternate technique.

System Programmer Response: None.

Source Data Set: UilSarProc.c, UilSarVal.c

Procedure Name: various

EZYXU19E *string value was found when string value was expected*

Explanation: The indicated value is not of the specific type required by UIL in this context.

System Action: The compiler continues.

User or Operator Response: Check the definition of the function or clause.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: various

EZYXU20W *The string string is not supported for the string object*

Explanation: Each widget or gadget supports a specific set of parameters, reasons, and children. The particular parameter, reason, or child you specified is not supported for this widget or gadget.

System Action: The compiler continues.

User or Operator Response: If a widget creation function accepts a parameter that UIL rejects, it does not necessarily mean that the UIL compiler is in error. Widget creation functions ignore parameters that they do not

support without notifying you that the parameter is being ignored.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU21I This *string string* supersedes a previous definition in this *string string*

Explanation: A parameter or call-back list has either a duplicate parameter or duplicate reason.

System Action: The compiler continues.

User or Operator Response: This is not necessarily an error. The compiler is alerting you to make sure that you intend to override the value of a prior parameter. This informational message can be suppressed using the -I option.

System Programmer Response: None.

Source Data Set: UilSemVal.c, UilSarMod.c

Procedure Name: various

EZYXU22E The name *string* previously defined as *string*

Explanation: The name marked by the message was used in a previous declaration. UIL requires that the names of all widgets declared within a module be unique.

System Action: The compiler continues.

User or Operator Response: Check for a misspelling. If the module is case-sensitive, the spellings of names in declarations and in references must match exactly.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sem_dcl_name

EZYXU23E The value used in this context must be private

Explanation: A private value is one that is not imported or exported. In the context marked by the message, only a private value is legal. Situations where this message is issued include defining one value in terms of another, and defining parameters in terms of functions. In general, a value must be private when the compiler must know the value at compilation time. Exported values are disallowed in these context, even though a value is present, because that value could be overridden at run time.

System Action: The compiler continues.

User or Operator Response: Change the value to private.

System Programmer Response: None.

Source Data Set: UilSarVal.c, UilSemVal.c

Procedure Name: various

EZYXU24E The procedure *string* was previously declared with *number* arguments

Explanation: The declaration of the marked procedure specified a different number of parameters than are present in this procedure reference.

System Action: The compiler continues.

User or Operator Response: Check that you are calling the correct function. If you intend to call the procedure with a varying number of parameters, omit the argument list in the procedure declaration.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_procref_entry

EZYXU25E *string value was found. The argument to procedure *string* must be *string* value*

Explanation: The declaration of the marked procedure specified a different type of parameters than is present in this procedure reference.

System Action: The compiler continues.

User or Operator Response: Check that you are passing the correct parameter to the correct function. If you intend to call the procedure with varying parameter types, declare the procedure specifying 'any' as the type of the parameter.

System Programmer Response: None.

Source Data Set: various

Procedure Name: various

EZYXU26E *string string was found when string string was expected*

Explanation: Most parameters take values of a specific type. The value specified is not correct for this procedure.

System Action: The compiler continues.

User or Operator Response: The message indicates the expected type of parameter. Check that you have specified the intended value and that you specified the correct parameter.

System Programmer Response: None.

Source Data Set: UilSarMod.c, UilSarObj.c

Procedure Name: various

EZYXU27E *string string was never defined*

Explanation: Certain UIL objects, such as gadgets and widgets, can be referred to before they are defined. The marked widget is such an object, however the compiler never found the widget declaration.

System Action: The compiler continues.

User or Operator Response: Check for misspelling. If the module is case-sensitive, the spellings of names in declarations and in references must match exactly.

System Programmer Response: None.

Source Data Set: UilP2Reslv.c

Procedure Name: sem_resolve_forward_ref

EZYXU28E *string string was already specified*

Explanation: A widget or gadget declaration can have, at most, one argument list, one call-back list, and one controls list.

System Action: The compiler continues.

User or Operator Response: If you want to specify multiple lists of arguments, controls and call-backs, you can do so within one list. For example: arguments (argument_list1; argument_list2;).

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_save_feature

EZYXU29E *string item is not allowed in string string*

Explanation: The indicated list item is not of the type required by the list. Argument lists must contain argument entries, call-back lists must contain call-back entries, control lists must contain control entries and procedure lists must contain procedure entries.

System Action: The compiler continues.

User or Operator Response: Check the syntax for the type of list entry that is required in this context and change the indicated list item.

System Programmer Response: None.

Source Data Set: UilSarObj.c, UilSemVal.c

Procedure Name: various

EZYXU30S **The compilation was terminated. Fix the previous errors**

Explanation: Errors encountered during the compilation have caused the compiler to stop processing.

System Action: The compiler ends.

User or Operator Response: Fix the errors already diagnosed by the compiler and re-compile.

System Programmer Response: None.

Source Data Set: UilDiags.c

Procedure Name: diag_issue_diagnostic

EZYXU31S **An internal error occurred. Submit a defect report.**

Explanation: The compiler diagnosed an internal error.

System Action: The compiler ends.

User or Operator Response: Get a listing and identify where the error is being issued. Try to correct any faulty syntax in this area. If you are unable to prevent this error, submit a software problem report.

System Programmer Response: None.

Source Data Set: UilDiags.c

Procedure Name: diag_issue_diagnostic

EZYXU33E *string was missing following "string" option*

Explanation: You used a command line option that requires a parameter and you did not provide that parameter.

System Action: The compiler continues.

User or Operator Response: Omit the option or provide the parameter.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU34S **An error occurred opening the listing file: *filename***

Explanation: The compiler could not create the listing file noted in the message.

System Action: The compiler ends.

User or Operator Response: Check that you have write access to the directory you specified to hold the listing file.

System Programmer Response: None.

Source Data Set: UilLstLst.c

EZYXU35S • EZYXU40S

Procedure Name: lst_open_listing

EZYXU35S An error occurred writing to the listing file: *filename*

Explanation: The compiler could not write a line into the listing file noted in the message.

System Action: The compiler ends.

User or Operator Response: Check to see that there is adequate space in the disk specified to hold the listing file.

System Programmer Response: None.

Source Data Set: UilLstLst.c

Procedure Name: lst_output_line

EZYXU36E An invalid module structure was detected. Check the UIL module syntax

Explanation: The structure of the UIL module is incorrect.

System Action: The compiler continues.

User or Operator Response: If there are any syntax errors reported, correct them and re-compile.

System Programmer Response: None.

Source Data Set: UilMain.c

Procedure Name: common_main

EZYXU37S Too many source files are open: *number*

Explanation: The compiler has a fixed limit for the number of source and include files that it can process. This number is reported in the message.

System Action: The compiler ends.

User or Operator Response: Use fewer include files.

System Programmer Response: None.

Source Data Set: UilSrcSrc.c

Procedure Name: src_open_file

EZYXU39I errors: *number* warnings: *number* informationals: *number*

Explanation: This message lists a summary of the diagnostics issued by the compiler and is displayed only when diagnostics have been issued.

System Action: The compiler continues.

User or Operator Response: Correct the problems reported. You can use the -I option to suppress informational and warning diagnostics that you have determined to be harmless.

System Programmer Response: None.

Source Data Set: UilDiags.c

Procedure Name: diag_issue_summary

EZYXU40S An error occurred opening the UID file: *filename*

Explanation: The compiler could not create the UID file noted in the message. A UID file holds the compiled user-interface specification.

System Action: The compiler ends.

User or Operator Response: Check that you have write access to the directory you specified to hold the UID file.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: sem_output_uid_file

EZYXU41I No UID file was produced

Explanation: If the compiler reported recoverable or nonrecoverable errors, no UID file is produced. The message informs you that the compiler did not produce a UID file.

System Action: The compiler continues.

User or Operator Response: Fix the problems reported by the compiler.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: sem_output_uid_file

EZYXU42E The creation procedure is not supported by the *string* widget

Explanation: You specified a creation procedure for a toolkit widget. You can specify a creation procedure only for a user-defined widget.

System Action: The compiler continues.

User or Operator Response: Remove the procedure clause following the widget type.

System Programmer Response: None.

Source Data Set: various

Procedure Name: various

EZYXU43E The creation procedure is not allowed in a *string* widget reference

Explanation: You specified a creation procedure when referencing a widget. You can specify a creation procedure only when you declare the widget.

System Action: The compiler continues.

User or Operator Response: Remove the procedure clause following the object type.

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_verify_object

EZYXU44E The creation procedure is required in a *string* widget declaration

Explanation: When defining a user-defined widget, you must specify the name of the creation function for creating an instance of this widget.

System Action: The compiler continues.

User or Operator Response: Insert a procedure clause following the widget type in the widget declaration. You also need to declare the creation procedure using a procedure declaration.

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_verify_object

EZYXU45W A NULL character in a string is not supported

Explanation: You have created a string that has an embedded null character. Strings are represented in a UID file and in many toolkit data structures as null-terminated strings. So, although the embedded nulls will be placed in the UID file, toolkit functions may interpret an imbedded null as the terminator for the string.

System Action: The compiler continues.

User or Operator Response: Be careful when using embedded nulls.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: yyparse

EZYXU46E Widget *string* is part of a circular definition

Explanation: The indicated object is referenced as a descendant of itself, either within its own definition or within the definition of one of the widgets in the widget tree that the object controls.

System Action: The compiler continues.

User or Operator Response: Change the definition of the indicated widget so that it is not a descendant of itself.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_callback_entry

EZYXU47S No source file was specified

Explanation: No source file was specified to compile.

System Action: The compiler ends.

User or Operator Response: Specify the name of a UIL specification file to compile.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU48W *string string* supports only a single *string string*

Explanation: You have specified a particular clause more than once in a context where that clause can only occur once. For example, the version clause in the module can only occur once.

System Action: The compiler continues.

User or Operator Response: Choose the correct clause and delete the others.

System Programmer Response: None.

Source Data Set: UilSarMod.c

Procedure Name: various

EZYXU49W *string* widget supports only a single control

Explanation: The specified widget may only have one entry in its controls list.

System Action: The compiler continues.

User or Operator Response: Change the control list to have only one entry.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: out_emit_widget

EZYXU51E Place the names clause before other module clauses

Explanation: The case-sensitive clause, if specified, must be the first clause following the name of the module. You have inserted another module clause before this clause.

System Action: The compiler continues.

User or Operator Response: Reorder the module clauses so that the case-sensitivity clause is first.

System Programmer Response: None.

Source Data Set: UilSarMod.c

Procedure Name: sar_process_module_sensitivity

EZYXU52E The color letter string must be a single character

Explanation: The string associated with each color in a color table must hold exactly one character. You have specified a string with either fewer or more characters.

System Action: The compiler continues.

User or Operator Response: Use a single character to represent each color in a color table.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sar_make_color_item

EZYXU53E The color letter was used for prior color in this table

Explanation: Each of the letters used to represent a color in a color table must be unique. If not, that letter in an icon would represent more than one color. The letter marked has been assigned to more than one color.

System Action: The compiler continues.

User or Operator Response: Use a single character to represent each color in a color table.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sar_make_color_item

EZYXU54E Row *number* must have same width as row 1

Explanation: The icons supported by UIL are rectangular. As a result, each of the strings used to represent a row of pixels in an icon must have the same length. The specified row does not have the same length as the first row.

System Action: The compiler continues.

User or Operator Response: Use a single character to represent each color in a color table.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sar_make_icon

EZYXU55E row *number*, column *number*: letter *letter* is not in the color table

Explanation: You have specified a color to be used in an icon that is not in the color table for that icon. The invalid color is identified in the message by displaying the letter used to represent it.

System Action: The compiler continues.

User or Operator Response: Either add the color to the color table for that icon, or use a character representing a color in the color table.

EZYXU56E • EZYXU62W

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_evaluate_value_expr

EZYXU56E There are too many *string* in *string*. The limit is *number*

Explanation: You exceeded a compiler limit such as the number of fonts in a font table or the number of strings in translation table. The message indicates the limit imposed by the compiler.

System Action: The compiler continues.

User or Operator Response: Restructure your UIL module.

System Programmer Response: None.

Source Data Set: UilCmd.c, UilSarVal.c

Procedure Name: various

EZYXU58W The *string* gadget is not supported. The *string* widget will be used instead

Explanation: The indicated widget type does not support a gadget variant, only a widget variant is supported for this widget type. The UIL compiler ignores the gadget indication and creates widgets of this widget type.

System Action: The compiler continues.

User or Operator Response: Restructure your UIL module.

System Programmer Response: None.

Source Data Set: UilSarMod.c, UilSarObj.c

Procedure Name: various

EZYXU59E The *string* type is not valid for *string*

Explanation: The indicated operand is not of a type that is supported by this operator.

System Action: The compiler continues.

User or Operator Response: Check the definition of the operator and make sure the type of the operand you specify is supported by the operator.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: validate_arg

EZYXU61W The *string* constraint is not supported for the *string* *string*

Explanation: You have specified a constraint that does not exist or is not appropriate for the constrained object.

System Action: The compiler continues.

User or Operator Response: Check for spelling errors. Check that the constraint is appropriate.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_constraint_entry

EZYXU62W Too many *string* options were detected, the limit is *number*

Explanation: You have specified more directories (using command line options) than the UIL compiler can process.

System Action: The compiler continues.

User or Operator Response: Reduce the number of directories specified by command line options.

System Programmer Response: None.

Source Data Set: UilCmd.c

Procedure Name: cmd_decode_command_line

EZYXU63W An error occurred while closing the source file: *filename*

Explanation: The UIL compiler was not able to properly close a source file.

System Action: The compiler continues.

User or Operator Response: Check the permissions of the file and its parent directory.

System Programmer Response: None.

Source Data Set: UilSrcSrc.c

Procedure Name: Uil_src_cleanup_source

EZYXU64E The *string* value is circularly defined

Explanation: You attempted to declare a value using an expression that contains the value you are declaring.

System Action: The compiler continues.

User or Operator Response: Remove the circular reference from the value expression.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU65W The *string* built-in name was overridden

Explanation: You have declared the meaning of a nonreserved keyword again. This is permitted, but generates a warning. You will not be able to use the UIL-supplied function for the keyword after it is overridden.

System Action: The compiler continues.

User or Operator Response: Make sure that you intended to override the built-in keyword. Also make sure that you no longer need the UIL-supplied function for that keyword.

System Programmer Response: None.

Source Data Set: UilSarVal.c

Procedure Name: sem_dcl_name

EZYXU66W The *string* argument does not support enumerated values

Explanation: You used the displayed argument along with a type that the argument does not support.

System Action: The compiler continues.

User or Operator Response: Correct the argument function. Use only supported types for the argument.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_argument_enumset

EZYXU67W The *string* argument does not support the *string* enumerated value

Explanation: You used the displayed argument along with an inappropriate value. Review the allowed values for the resources supported by the widget.

System Action: The compiler continues.

User or Operator Response: Correct the argument function. Use only supported values for the argument.

EZYXU68E • EZYXU71S

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_validate_argument_enumset

EZYXU68E The environment variable, \$LANG, contains an unknown character set

Explanation: The \$LANG environment variable contains an unknown character set. The UIL compiler does not have a default definition for the character set, and you have not supplied a definition using the CHARACTER_SET function.

System Action: The compiler continues.

User or Operator Response: Make sure the character set is spelled correctly. If it is a user-defined character set, make sure you have defined it within a CHARACTER_SET function.

System Programmer Response: None.

Source Data Set: UilLexAna.c

Procedure Name: lex_initialize_analyzer

EZYXU69E The *string* object's controls hierarchy contains a reference to itself

Explanation: You have referred to a widget within its own controls list. Widgets cannot act as their own controlled descendants.

System Action: The compiler continues.

User or Operator Response: Remove the circular reference from the widget controls list.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU70S The value *string* is too large for context buffer

Explanation: The UIL compiler was unable to allocate an internal buffer while attempting to parse the specified value.

System Action: The compiler ends

User or Operator Response: If possible, reduce the complexity of the value expression.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: out_emit_value

EZYXU71S Forward referencing is not allowed for *string*

Explanation: You have referenced an object before defining it.

System Action: The compiler ends

User or Operator Response: Restructure the module so the object is defined before being referenced.

System Programmer Response: None.

Source Data Set: UilSarMod.c

Procedure Name: sar_process_module_version

EZYXU72E *string type cannot be converted to string type*

Explanation: You have defined an expression containing two types of operands that cannot be converted into one another (for example, integer and Pixmap) or that require an explicit conversion function (for example, Boolean and Floating Point).

System Action: The compiler continues

User or Operator Response: Make sure that the expression is correct and that the required conversion functions are used.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU73E *string is invalid*

Explanation: You defined an argument function using an argument that UIL does not allow.

System Action: The compiler continues

User or Operator Response: Make sure the argument function is correct.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: various

EZYXU74S *An error occurred while reading the binary database*

Explanation: The binary database file could not be successfully read.

System Action: The compiler ends

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c

Procedure Name: various

EZYXU75S *The binary database was compiled with a future version*

Explanation: The binary database was compiled by a UIL version later than the compiler version being used.

System Action: The compiler ends

User or Operator Response: Use a binary database compiled with the current UIL compiler. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c

Procedure Name: db_incorporate

EZYXU76S *An error occurred while opening the database file: filename*

Explanation: The binary database file could not be opened.

System Action: The compiler ends

User or Operator Response: Verify that the file exists and has the proper permissions. Invoke the compiler again.

System Programmer Response: None.

Source Data Set: UilDB.c

Procedure Name: db_open_file

EZYXU77S An error occurred while writing to the UID file: *filename*

Explanation: The compiler could not write a line into the UID file noted in the message.

System Action: The compiler ends

User or Operator Response: Check to see that there is adequate space in the disk specified to hold the UID file.

System Programmer Response: None.

Source Data Set: UilP2Out.c

Procedure Name: various

EZYXU79S Invalid include file name

Explanation: The value_entry structure is defined as a type other than char_8, when char_8 is required.

System Action: The compiler ends

User or Operator Response: Ensure usage of the correct structure type.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: UilSarInc.c

Procedure Name: sar_include_file()

EZYXU80W Incompatible unit types for arithmetic operation

Explanation: While performing an arithmetic operation with multiple arguments, the compiler found that the arguments were of incompatible types. For example, the compiler was expecting to perform an AND operation on two integer variables, and found that one variable was a floating point.

System Action: The compiler ends

User or Operator Response: Verify that the UID file has Integer and FloatingPoint Values in the correct order or place.

System Programmer Response: None.

Source Data Set: UilSemVal.c

Procedure Name: sem_evaluate_value_expr()

EZYXU81I *string* used as charset name; *string* used as charset component

Explanation: During NAME to CHARSET_NAME conversion, the specified charset could not be resolved. The default charset was used.

System Action: None.

User or Operator Response: If the default charset is not acceptable, verify that the correct charset is accessible and correctly labeled.

System Programmer Response: None.

Source Data Set: UilSarComp.c

Procedure Name: sar_make_fallback_charset()

EZYXU82E *string string* already specified for this *string string*

Explanation: A widget or gadget declaration can have no more than one argument list, one call-back list, and one control list.

System Action: The compiler continues.

User or Operator Response: If you want to specify multiple lists of arguments, controls and call-backs, you can do so within one list. For example: arguments (argument_list1; argument_list2;).

System Programmer Response: None.

Source Data Set: UilSarObj.c

Procedure Name: sar_save_feature()

EZYXU99E *string1 string2 string3*

Explanation: The Mrm function, *string1*, has detected an error. *string2* describes the error. *string3* states the status returned by the Mrm function.

System Action: None.

User or Operator Response: None.

System Programmer Response: Based on the error description, review the application code and the UIL module. Verify that the correct UID module is being accessed. In most cases, this is an application programming error. If the error description states that this is an internal error, submit a software problem report.

Source Data Set: Mrmerror.c

Procedure Name: Urm__UT_Error

EZYXW01E Xlib: connection to string refused to server

Explanation: The server identified by string has refused the connection.

System Action: The application is ended.

User or Operator Response: Issue the xhost command at the server to add the host address.

System Programmer Response: None.

Source Data Set: OpenDis.c

Procedure Name: XOpenDisplay

EZYXW02E Xlib: client uses different protocol version (number) than server (number)!

Explanation: There is a mismatch between the version of the X Window System protocol used by the client library and that used by the server (string2)

System Action: The application is ended.

User or Operator Response: Use a server which is compatible with the client library.

System Programmer Response: None.

Source Data Set: OpenDis.c

Procedure Name: XOpenDisplay

EZYXW03E Error parsing argument "string1" (string2); string3

Explanation: The Resource Manager has been passed an invalid argument. String1 is the option name; string2 is the value supplied; string3 is an explanatory comment.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: ParseCmd.c

Procedure Name: _XReportParseError

EZYXW04E X connection to string broken (explicit kill or server shutdown).

Explanation: The server has been shut down or the client window has been destroyed. This message may be generated when a window is closed by the user and the application does not recognize what has happened.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XDefaultIOError

EZYXW05E XIO: fatal IO error number1 (string1) on X Server "string2" after number2 requests (number3 known processed) with number4 events remaining.

Explanation: The connection to the server has been broken. The reported error is number1 with a descriptive string string1. String2 identifies the server being used.

System Action: The application is ended.

User or Operator Response: Try to restart the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XDefaultIOError

EZYXW06E Xlib: extension "string1" string2 on display "string2".

Explanation: An error has been detected by an X Window System extension. String1 is the name of the extension; string2 is a description of the error; String3 is the address of the server.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: extutil.c

Procedure Name: _default_exterror

EZYXW07W Warning: Current locale is not supported by Xlib

Explanation: The current locale is not supported by Xlib. The server code set is set to ISO8859-1.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Conv.c

Procedure Name: init_iconv

EZYXW08W Warning: Unable to convert from string1 to string2.

Explanation: Conversion tables do not exist for converting from code set string1 to codeset string2. The default conversions are used.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Conv.c

Procedure Name: init_iconv

EZYXW09I Unable to open message catalog: X11R6.cat

Explanation: The message catalog for X Window System messages could not be opened.

System Action: The application continues.

User or Operator Response: Verify that the NLSPATH environment variable is set to the correct value.

System Programmer Response: None.

Source Data Set: OpenDis.c, Xt/Initialize.c

Procedure Name: XOpenDisplay,XtToolitInitialize

EZYXW10E X Error of failed request: string Major op code of failed request: number string (additional lines depending on X Error) Serial number of failed request: number Current serial number in output stream: number

Explanation: This message is displayed by the default Xlib error handler. An X Window System protocol error has been detected. The type of error detected is described by string. The failed request Major op code is displayed as a number and as a descriptive string. This is followed by one or more of the following lines, depending on the type of error: ResourceID in failed request: hexadecimal number AtomID in failed request: hexadecimal number value in failed request: hexadecimal number Minor code of failed request: number. These lines are followed by the serial number of the failed request and the current serial number.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: XlibInt.c

Procedure Name: _XPrintDefaultError

EZYXW11E XtlibError: string1 string2 string3 string4

Explanation: An Xtlib function has detected an error. String1 is the name of the function reporting the error, string2 is the error type, string3 is the error class, and string4 is a descriptive string.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: XtMsgCat.c

Procedure Name: XtErrorMsgCat

EZYXW12W XtlibWarning: string1 string2 string3 string4

Explanation: An Xtlib function has detected a recoverable error. String1 is the name of the function reporting the error, string2 is the error type, string3 is the error class, and string4 is a descriptive string.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: XtMsgCat.c

EZYXW13W • EZYXW17W

Procedure Name: XtWarningMsgCat

EZYXW13W Xtlib: locale not supported by C library, locale unchanged.

Explanation: An attempt was made to set the current locale to a value not supported by the C library.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Initialize.c

Procedure Name: _XtDefaultLanguageProc

EZYXW14W Xtlib: locale not supported by Xlib, locale set to C

Explanation: An attempt was made to set the current locale to a value not supported by Xlib.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Initialize.c

Procedure Name: _XtDefaultLanguageProc

EZYXW15W Xtlib: X Locale modifiers not supported, using default

Explanation: The locale modifiers could not be set.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: Initialize.c

Procedure Name: _XtDefaultLanguageProc

EZYXW16W XtVaTypedArg is not valid in XtVaSetSubvalues()

Explanation: XtVaSetSubvalues has been called with typed arguments.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: VarCreate.c

Procedure Name: XtVaSetSubvalues

EZYXW17W XtVaTypedArg is an invalid argument to XtVaGetSubvalues()

Explanation: XtGetSubvalues has been called with typed arguments.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: VarGet.c

Procedure Name: XtVaGetSubvalues

EZYXW18W String to BackingStore conversion needs no extra arguments

Explanation: The string to backing store conversion has been called with an extra argument.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xmu/StrToBS.c

Procedure Name: XmuCvtStringToBackingStore

EZYXW19W Xawlib: Too many parameters passed to highlight action table.

Explanation: Extra parameters were passed in a call to Highlight().

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Command.c

Procedure Name: Highlight

EZYXW20W List Widget: Unknown geometry return.

Explanation: A call to ChangeSize contained an unknown geometry request.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/List.c

Procedure Name: ChangeSize

EZYXW21W List Widget: Size changed when it shouldn't have when resizing.

Explanation: An attempt was made to resize the List widget inappropriately.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/List.c

Procedure Name: Resize

EZYXW22E MenuButton: Could not find menu widget named string.

Explanation: An incorrect name was passed for a menu widget.

System Action: The application continues.

User or Operator Response: None.

EZYXW23W • EZYXW26W

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: MenuButton.c

Procedure Name: PopupMenu

EZYXW23W MultiSrc: The XtNuseStringInPlace resources may not be changed.

Explanation: An attempt was made to change the XtNuseStringInPlace resource.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/MultiSrc.c

Procedure Name: SetValues

EZYXW24E string1: unable to allocate number bytes for widget name

Explanation: Unable to allocate the required number of bytes for a widget name. String1 is the name of the routine allocating the storage.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xmu/WidgetNode.c

Procedure Name: XmuWnInitializeNodes

EZYXW25E String1: unable to calloc number1 number2 byte widget node ptrs

Explanation: Unable to allocate number1 number2 byte areas of storage for use as widget node pointers.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xmu/WidgetNode.c

Procedure Name: XmuWnFetchResources

EZYXW26W XawTextWidget: An attempt was made to insert an illegal selection.

Explanation: An attempt was made to insert an illegal selection into an Xaw Text widget.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: TextAction.c

Procedure Name: StartAction

EZYXW27W Xaw MultiSrc Object: possible memory leak in FreeAllPieces().

Explanation: A possible memory leak has been detected in FreeAllPieces().

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: MultiSrc.c

Procedure Name: FreeAllPieces

EZYXW28E Paned GripAction(); 1st parameter invalid.

Explanation: The first parameter passed to HandleGrip is invalid.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Paned.c

Procedure Name: HandleGrip

EZYXW29W Scrollbar Widget: Could not get geometry of thumb pixmap.

Explanation: The scrollbar widget could not get the geometry of the thumb pixmap.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Scrollbar.c

Procedure Name: CreateGC

EZYXW30W string Widget: The Simple Widget class method 'change_sensitive' must be defined or inherited.

Explanation: The widget named string does not have the Simple Widget class method 'change_sensitive' defined or inherited.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Simple.c

Procedure Name: ClassPartInitialize

EZYXW31E Paned GripAction has been passed incorrect parameters.

Explanation: HandleGrip has been passed incorrect parameters.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Paned.c

Procedure Name: HandleGrip

EZYXW32E Unknown event type in GetEventEntry().

Explanation: An unknown event type has been detected.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: SimpleMenu.c

Procedure Name: GetEventEntry

EZYXW33W No Dynamic class change of the SimpleMenu Label.

Explanation: An attempt was made to change the class of the SimpleMenu label.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: GetValues

EZYXW34W Xaw - SimpleMenuWidget: position menu action expects only one parameter which is the name of the menu.

Explanation: There should be only one parameter passed to PositionMenuAction.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: PositionMenuAction

EZYXW35W Xaw - SimpleMenuWidget: could not find menu named: string

Explanation: The menu widget named string could not be found.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: PositionMenuAction

EZYXW36W Xaw Simple Menu Widget: label string is NULL or label already exists, no label is being created.

Explanation: The label string is NULL or the label already exists for a SimpleMenu widget.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: CreateLabel

EZYXW37W Xaw Simple Menu Widget: Could not find location of mouse pointer.

Explanation: The position of the mouse pointer could not be determined.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SimpleMenu.c

Procedure Name: PositionMenu

EZYXW38E Xaw SmeBSB Object: Could not get Left Bitmap geometry information for menu entry "string"

Explanation: The Left Bitmap geometry information for the menu entry named string could not be obtained.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW39E Xaw SmeBSB Object: Left Bitmap of entry "string" is not one bit deep.

Explanation: The bitmap specified for entry string is not one bit deep.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW40E Xaw SmeBSB Object: Could not get Right Bitmap geometry information for menu entry "string".

Explanation: The Right Bitmap geometry information for the menu entry named string could not be obtained.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW41E Xaw SmeBSB Object: Right Bitmap of entry "string" is not one bit deep.

Explanation: The bitmap specified for entry string is not one bit deep.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/SmeBSB.c

Procedure Name: GetBitmapInfo

EZYXW42W Xaw Text Widget string: Vertical scrolling not allowed with height resize. Vertical scrolling has been DEACTIVATED.

Explanation: An attempt was made to specify vertical scrolling with height resizing for Text widget string.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: Initialize

EZYXW43W Xaw Text Widget string: Horizontal scrolling not allowed with wrapping active. Horizontal scrolling has been DEACTIVATED.

Explanation: An attempt was made to specify horizontal scrolling when wrapping was active for Text widget string.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: Initialize

EZYXW44W Xaw Text Widget string: Horizontal scrolling not allowed with width resize. Horizontal scrolling has been DEACTIVATED.

Explanation: An attempt was made to specify horizontal scrolling and width resize for Text widget string.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: Initialize

EZYXW45W Xaw Text Widget: empty selection array.

Explanation: The selection array passed to DoSelection is empty.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: DoSelection

EZYXW46E Xaw Text Widget: multiply() takes exactly one argument.

Explanation: An incorrect number or arguments was passed to multiply().

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/TextAction.c

Procedure Name: Multiply

EZYXW47E Xaw Text Widget: multiply() argument must be a number greater than zero, or 'Reset'.

Explanation: The argument to multiply must be a number greater than zero, or Reset.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/TextAction.c

Procedure Name: Multiply

EZYXW48W string This action must have only one or two parameters.

Explanation: The action, string, must have only one or two parameters.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/TextPop.c

Procedure Name: _XawTextSearch

EZYXW49W string The first parameter must be either backward or forward.

Explanation: The first parameter for the Search routine of the Text Widget is not either 'backward' or 'forward'.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/TextPop.c

EZYXW50W • EZYXW54W

Procedure Name: Search

EZYXW50W Toggle Widget Error - Attempting to create a new toggle group when one already exists.

Explanation: An attempt was made to create a new toggle group when one already exists.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Toggle.c

Procedure Name: CreateRadioGroup

EZYXW51E Aborting, due to errors resolving bindings in the Toggle widget.

Explanation: There is an error in the Toggle widget's action table.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Toggle.c

Procedure Name: ClassInit

EZYXW52W we can not open any input method

Explanation: Xaw is unable to open any input method.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/XawIm.c

Procedure Name: OpenIM

EZYXW53W Xaw: input method doesn't support any style

Explanation: The current input method does not support any style.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/XawIm.c

Procedure Name: OpenIM

EZYXW54W Xaw: input method doesn't support my input style.

Explanation: The input method being opened does not specify any supported style.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/XawIm.c

Procedure Name: OpenIM

EZYXW55E XSMP error: Offending minor code = number (string) Offending sequence number = number Error class = string Severity = string (may be followed by:) BadValue Offset = number BadValue Length = number BadValue = number

Explanation: The session manager client routines have detected an error. The Offending minor code is displayed as a number and an explanatory string. The Error class and severity are displayed as strings. If the Error class is BadValue, the offset, length, and the bad value are displayed.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: SM/sm_error.c

Procedure Name: _SmcDefaultErrorHandler

EZYXW56E ICE error: Offending minor code = number (string) Offending sequence number = number Error class = string Severity = string (may be followed by one or more of the following:) BadValue Offset = string BadValue Length = string BadValue = string Major opcode : number Reason : string Protocol name : string

Explanation: ICE has detected an error. Information about the error is displayed.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: ICE/error.c

Procedure Name: _IceDefaultErrorHandler

EZYXW57E X Error: string

Explanation: An X Protocol error has been detected. String displays a description of the error.

System Action: The application is ended.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xmu/DefErrMsg.c

Procedure Name: XmuPrintDefaultErrorMessage

EZYXW58W XSMP error: Offending minor code = number (string) Offending sequence number = number Error class = string Severity = string (may be followed by:) BadValue Offset = number BadValue Length = number BadValue = number

Explanation: The session manager server routines have detected an error. The Offending minor code is displayed as a number and an explanatory string. The Error class and severity are displayed as strings. If the Error class is BadValue, the offset, length, and the bad value are displayed.

System Action: The application continues.

EZYXW59W • EZYXW62I

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: SM/sm_error.c

Procedure Name: _SmcDefaultErrorHandler

EZYXW59W Xaw Text widget string: empty copy queue.

Explanation: An attempt was made to copy with an empty copy queue.

System Action: The application continues.

User or Operator Response: None.

System Programmer Response: Use your X Window System/Motif documentation to correct this application programming error.

Source Data Set: Xaw/Text.c

Procedure Name: PopCopyQueue

EZYXW60I Xlib: resource ID allocation space exhausted!

Explanation: The X Window System resource ID space has been exhausted.

System Action: The application continues.

User or Operator Response: None. value.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XAllocID

EZYXW61I Xlib: sequence lost (number > number) in reply type number!

Explanation: A sequence number has been received which is less than the last received sequence number.

System Action: The application continues.

User or Operator Response: Data may have been lost, consider restarting the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XSetLastRequestRead

EZYXW62I Xlib: unexpected async reply (sequence number)!

Explanation: An asynchronous reply has been received when None. was expected.

System Action: The application continues.

User or Operator Response: Data may have been lost, consider restarting the application.

System Programmer Response: None.

Source Data Set: XlibInt.c

Procedure Name: _XAsyncReply

EZYXW6eI Xlib: unexpected connection setup reply from server, type 'nbr'.

Explanation: An unexpected reply has been received the server. This could be an unexpected authenticate reply.

System Action: The application is ended.

User or Operator Response: Verify that you are meeting the authentication requirements of the server you are connecting to. Try again.

System Programmer Response: None.

Source Data Set: OpenDis.c

Procedure Name: XOpenDisplay

EZYXW63E Could not initialize the PEX extension on the specified display.

Explanation: The PEX server on the display could not be properly initialized.

System Action: The application is ended.

User or Operator Response: Verify that the PEX extension is active for the server.

System Programmer Response: None.

Source Data Set: PEX5/pl_startup.c

Procedure Name: PEXInitialize

EZYXW64E Could not allocate memory for PEXlib internal usage.

Explanation: PEXlib could not allocate memory.

System Action: The application is ended.

User or Operator Response: Increase virtual memory size.

System Programmer Response: None.

Source Data Set: PEX5/pl_startup.c

Procedure Name: PEXInitialize

EZYXW65E Implicit call to PEXGetEnumTypeInfo by PEXInitialize failed.

Explanation: PEXlib could not initialize properly.

System Action: The application is ended.

User or Operator Response: Verify that the PEX extension is active for the server.

System Programmer Response: None.

Source Data Set: PEX5/pl_startup.c

Procedure Name: PEXInitialize

EZYXW66E No floating point formats supported by server.

Explanation: The PEX server extension does not support any floating point formats supported by the client.

System Action: The application is ended.

User or Operator Response: Use a server which supports the required floating point format.

System Programmer Response: None.

Source Data Set: PEX5/pl_startup.c

Procedure Name: PEXInitialize

EZYXW67E Could not get PEX extension information.

Explanation: Information about the PEX server extension could not be obtained.

System Action: The application is ended.

User or Operator Response: Verify that the PEX server extension is active.

System Programmer Response: None.

Source Data Set: PEX5/pl_startup.c

Procedure Name: PEXInitialize

EZYXW68E Client speaks PEX num.num; Server speaks PEX num.num

Explanation: Attempting to use incompatible PEX client and server.

System Action: The application is ended.

User or Operator Response: Verify that the correct PEX server extension is active.

System Programmer Response: None.

Source Data Set: PEX5/pl_startup.c

Procedure Name: PEXInitialize

EZYXW69I Opcode of failed output command: number. Number of output commands processed: number.

Explanation: Displays the PEX opcode which failed.

System Action: N/A

User or Operator Response: None.

System Programmer Response: Correct the application and try again.

Source Data Set: PEX5/pl_startup.c

Procedure Name: _PEXPrintOCError

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 - EZA4432	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	TCP/IP

Volume 2 ranges

Number Span	Component
EZB0600 - EZB0857	LPD
EZB0900 - EZB1100	LPT
EZB1200 - EZB1230	X Window System
EZB2000 - EZB2498	XNX25
EZB2500 - EZB2785	Retired — do not use
EZB3000 - EZB3566	DNS
EZB3825 - EZB4198	NCPROUTE
EZB6473	Initialization
EZB8801	FFST™
EZD0001 - EZD0050	Interface Layer
EZD0101 - EZD0799	Netstat
EZD0800 - EZD0801	Enterprise Extender
EZD0810 - EZD1160	IPSec
EZD1161 - EZD1162	DCAS configuration file
EZD1163 - EZD1214	DVIPA

Number Span	Component
EZD1215 - EZD1230	Multilevel Security Enhancements
EZD1231 - EZD1280	z/OS Load Balancing Advisor
EZD1281 - EZD1290	AT-TLS

Volume 3 ranges

Number Span	Component
EZY0002 - EZY0058	Packet Trace
EZY0620 - EZY0974	Network Print facility
EZY1020 - EZY1035	IMS Socket
EZY1120 - EZY1130	Sockets Extended
EZY1218 - EZY1371	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	REXEC
EZYRD01 - EZYRD39	REXECD
EZYRP35 - EZYRP72 (EZA4331 - EZA4335)	RPC
EZYRS01 - EZYRS66	RSHD
EZYTE01 - EZYTE87 EZYTO01 - EZYTO12 EZYTS02 - EZYTS18 EZYTU01 - EZYTU36 EZYTY01 - EZYTY14	z/OS UNIX Telnet
EZYXM01 - EZYXR42	Motif API
EZYXU01 - EZYXU99	Motif User Interface Language Compiler
EZYXW01 - EZYXW69	X Window System

Volume 4 ranges

Number Span	Component
EZZ0053 - EZZ0720	Configuration & Initialization
EZZ2351 - EZZ3089	ONETSTAT & NETSTAT
EZZ3105 - EZZ3133	OPING
EZZ3200 - EZZ3249	SNMP Subagent

Number Span	Component
EZZ3250 - EZZ3255	Operator Command
EZZ3300 - EZZ3344	OSNMP
EZZ4200 - EZZ4247	Initialization
EZZ4300 - EZZ4349	Interface Layer
EZZ4350	Initialization
EZZ4360 - EZZ4367	Storage Utilization
EZZ4820 - EZZ5027	ROUTED
EZZ6002 - EZZ6084	Telnet
EZZ6201 - EZZ6319	SNMP Agent
EZZ6351 - EZZ6388	OTRACERT
EZZ6452 - EZZ6705	DNS
EZZ7001 - EZZ7050	TFTP
EZZ7051 - EZZ7062	TIMED
EZZ7251 - EZZ7331	DHCP
EZZ7350 - EZZ7448	NSM
EZZ7450 - EZZ7452	Initialization
EZZ7475 - EZZ7478	Stack Configuration
EZZ7500 - EZZ7738	Sendmail & Popper
EZZ7800 - EZZ8111	OMPROUTE
EZZ8201 - EZZ8251	Policy Agent subagent
EZZ8252 - EZZ8260	D TCPIP,,SYSPLEX
EZZ8301 - EZZ8319	VIPA
EZZ8320 - EZZ8331	HOSTNAME, DOMAINNAME, DNSDOMAINNAME, HOST
EZZ8340 - EZZ8345	Misc. Command
EZZ8392 - EZZ8423	Trap Forwarder Daemon
EZZ8431 - EZZ8452	Policy Agent (PAGENT)
EZZ8453 - EZZ8460	Storage Display
EZZ8461 - EZZ8477	Sysplex Distributor
EZZ8495- EZZ8503	Intrusion Detection Services (SYSLOG)
EZZ8504 - EZZ8511	Intrusion Detection Services Statistics (TRMDSTAT)
EZZ8540	Intrusion Detection Services (SYSLOG)
EZZ8600 - EZZ8635	DCAS
EZZ8638 - EZZ8659	Intrusion Detection Services SYSLOG
EZZ8761 - EZZ8770	Intrusion Detection Services Console
EZZ8771 - EZZ8790	Policy Agent (PAGENT)
EZZ8791 - EZZ9290	DNS BIND 9

Number Span	Component
EZZ9291 - EZZ9304	System Resolver
EZZ9315	Initialization
EZZ9316 - EZZ9327	Intrusion Detection Services SYSLOG
EZZ9331 - EZZ9599	DNS BIND 9
EZZ9600 - EZZ9602	SNTP
EZZ9650 - EZZ9669	VIPA
EZZ9670 - EZZ9679	Sysplex Distributor
EZZ9680 - EZZ9724	DNS Bind 9
EZZ9780 - EZZ9784	Neighbor Discovery
EZZ9790 - EZZ9817	FTP
EZZ9891 - EZZ9993	Sendmail
SNM000 - SNM129	SNMPLIB

Appendix B. 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.

You can request RFCs through electronic mail, from the automated Network Information Center (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` or at:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Hard copies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available at the following Web address:
<http://www.rfc-editor.org/rfc.html>.

See "Internet drafts" on page 568 for draft RFCs implemented in this and previous Communications Server releases.

Many features of TCP/IP Services are based on the following RFCs:

RFC	Title and Author
-----	------------------

652	<i>Telnet output carriage-return disposition option</i> D. Crocker
653	<i>Telnet output horizontal tabstops option</i> D. Crocker
654	<i>Telnet output horizontal tab disposition option</i> D. Crocker
655	<i>Telnet output formfeed disposition option</i> D. Crocker
657	<i>Telnet output vertical tab disposition option</i> D. Crocker
658	<i>Telnet output linefeed disposition</i> D. Crocker
698	<i>Telnet extended ASCII option</i> T. Mock
726	<i>Remote Controlled Transmission and Echoing Telnet option</i> J. Postel, D. Crocker
727	<i>Telnet logout option</i> M.R. Crispin
732	<i>Telnet Data Entry Terminal option</i> J.D. Day
733	<i>Standard for the format of ARPA network text messages</i> D. Crocker, J. Vittal, K.T. Pogran, D.A. Henderson

	734	<i>SUPDUP Protocol</i> M.R. Crispin
	735	<i>Revised Telnet byte macro option</i> D. Crocker, R.H. Gumpertz
	736	<i>Telnet SUPDUP option</i> M.R. Crispin
	749	<i>Telnet SUPDUP—Output option</i> B. Greenberg
	765	<i>File Transfer Protocol specification</i> J. Postel
	768	<i>User Datagram Protocol</i> J. Postel
	779	<i>Telnet send-location option</i> E. Killian
	783	<i>TFTP Protocol (revision 2)</i> K.R. Sollins
	791	<i>Internet Protocol</i> J. Postel
	792	<i>Internet Control Message Protocol</i> J. Postel
	793	<i>Transmission Control Protocol</i> J. Postel
	820	<i>Assigned numbers</i> J. Postel
	821	<i>Simple Mail Transfer Protocol</i> J. Postel
	822	<i>Standard for the format of ARPA Internet text messages</i> D. Crocker
	823	<i>DARPA Internet gateway</i> R. Hinden, A. Sheltzer
	826	<i>Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware</i> D. Plummer
	854	<i>Telnet Protocol Specification</i> J. Postel, J. Reynolds
	855	<i>Telnet Option Specification</i> J. Postel, J. Reynolds
	856	<i>Telnet Binary Transmission</i> J. Postel, J. Reynolds
	857	<i>Telnet Echo Option</i> J. Postel, J. Reynolds
	858	<i>Telnet Suppress Go Ahead Option</i> J. Postel, J. Reynolds
	859	<i>Telnet Status Option</i> J. Postel, J. Reynolds
	860	<i>Telnet Timing Mark Option</i> J. Postel, J. Reynolds
	861	<i>Telnet Extended Options: List Option</i> J. Postel, J. Reynolds
	862	<i>Echo Protocol</i> J. Postel
	863	<i>Discard Protocol</i> J. Postel
	864	<i>Character Generator Protocol</i> J. Postel
	865	<i>Quote of the Day Protocol</i> J. Postel
	868	<i>Time Protocol</i> J. Postel, K. Harrenstien
	877	<i>Standard for the transmission of IP datagrams over public data networks</i> J.T. Korb
	883	<i>Domain names: Implementation specification</i> P.V. Mockapetris
	884	<i>Telnet terminal type option</i> M. Solomon, E. Wimmers
	885	<i>Telnet end of record option</i> J. Postel
	894	<i>Standard for the transmission of IP datagrams over Ethernet networks</i> C. Hornig
	896	<i>Congestion control in IP/TCP internetworks</i> J. Nagle

	903	<i>Reverse Address Resolution Protocol</i> R. Finlayson, T. Mann, J. Mogul, M. Theimer
	904	<i>Exterior Gateway Protocol formal specification</i> D. Mills
	919	<i>Broadcasting Internet Datagrams</i> J. Mogul
	922	<i>Broadcasting Internet datagrams in the presence of subnets</i> J. Mogul
	927	<i>TACACS user identification Telnet option</i> B.A. Anderson
	933	<i>Output marking Telnet option</i> S. Silverman
	946	<i>Telnet terminal location number option</i> R. Nedved
	950	<i>Internet Standard Subnetting Procedure</i> J. Mogul, J. Postel
	951	<i>Bootstrap Protocol</i> W.J. Croft, J. Gilmore
	952	<i>DoD Internet host table specification</i> K. Harrenstien, M. Stahl, E. Feinler
	959	<i>File Transfer Protocol</i> J. Postel, J.K. Reynolds
	961	<i>Official ARPA-Internet protocols</i> J.K. Reynolds, J. Postel
	974	<i>Mail routing and the domain system</i> C. Partridge
	1001	<i>Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods</i> NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force
	1002	<i>Protocol Standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications</i> NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force
	1006	<i>ISO transport services on top of the TCP: Version 3</i> M.T. Rose, D.E. Cass
	1009	<i>Requirements for Internet gateways</i> R. Braden, J. Postel
	1011	<i>Official Internet protocols</i> J. Reynolds, J. Postel
	1013	<i>X Window System Protocol, version 11: Alpha update April 1987</i> R. Scheifler
	1014	<i>XDR: External Data Representation standard</i> Sun Microsystems
	1027	<i>Using ARP to implement transparent subnet gateways</i> S. Carl-Mitchell, J. Quarterman
	1032	<i>Domain administrators guide</i> M. Stahl
	1033	<i>Domain administrators operations guide</i> M. Lottor
	1034	<i>Domain names—concepts and facilities</i> P.V. Mockapetris
	1035	<i>Domain names—implementation and specification</i> P.V. Mockapetris
	1038	<i>Draft revised IP security option</i> M. St. Johns
	1041	<i>Telnet 3270 regime option</i> Y. Rekhter
	1042	<i>Standard for the transmission of IP datagrams over IEEE 802 networks</i> J. Postel, J. Reynolds
	1043	<i>Telnet Data Entry Terminal option: DODIIS implementation</i> A. Yasuda, T. Thompson
	1044	<i>Internet Protocol on Network System's HYPERchannel: Protocol specification</i> K. Hardwick, J. Lekashman
	1053	<i>Telnet X.3 PAD option</i> S. Levy, T. Jacobson

- 1055 *Nonstandard for transmission of IP datagrams over serial lines: SLIP* J. Romkey
- 1057 *RPC: Remote Procedure Call Protocol Specification: Version 2* Sun Microsystems
- 1058 *Routing Information Protocol* C. Hedrick
- 1060 *Assigned numbers* J. Reynolds, J. Postel
- | 1067 *Simple Network Management Protocol* J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin
- |
- | 1071 *Computing the Internet checksum* R.T. Braden, D.A. Borman, C. Partridge
- | 1072 *TCP extensions for long-delay paths* V. Jacobson, R.T. Braden
- 1073 *Telnet window size option* D. Waitzman
- 1079 *Telnet terminal speed option* C. Hedrick
- | 1085 *ISO presentation services on top of TCP/IP based internets* M.T. Rose
- 1091 *Telnet terminal-type option* J. VanBokkelen
- 1094 *NFS: Network File System Protocol specification* Sun Microsystems
- 1096 *Telnet X display location option* G. Marcy
- 1101 *DNS encoding of network names and other types* P. Mockapetris
- | 1112 *Host extensions for IP multicasting* S.E. Deering
- | 1113 *Privacy enhancement for Internet electronic mail: Part I — message encipherment and authentication procedures* J. Linn
- |
- 1118 *Hitchhikers Guide to the Internet* E. Krol
- 1122 *Requirements for Internet Hosts—Communication Layers* R. Braden, Ed.
- 1123 *Requirements for Internet Hosts—Application and Support* R. Braden, Ed.
- | 1146 *TCP alternate checksum options* J. Zweig, C. Partridge
- 1155 *Structure and identification of management information for TCP/IP-based internets* M. Rose, K. McCloghrie
- 1156 *Management Information Base for network management of TCP/IP-based internets* K. McCloghrie, M. Rose
- 1157 *Simple Network Management Protocol (SNMP)* J. Case, M. Fedor, M. Schoffstall, J. Davin
- 1158 *Management Information Base for network management of TCP/IP-based internets: MIB-II* M. Rose
- | 1166 *Internet numbers* S. Kirkpatrick, M.K. Stahl, M. Recker
- 1179 *Line printer daemon protocol* L. McLaughlin
- 1180 *TCP/IP tutorial* T. Socolofsky, C. Kale
- | 1183 *New DNS RR Definitions* C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris
- |
- 1184 *Telnet Linemode Option* D. Borman
- | 1186 *MD4 Message Digest Algorithm* R.L. Rivest
- 1187 *Bulk Table Retrieval with the SNMP* M. Rose, K. McCloghrie, J. Davin
- 1188 *Proposed Standard for the Transmission of IP Datagrams over FDDI Networks* D. Katz

- 1190 *Experimental Internet Stream Protocol: Version 2 (ST-II)* C. Topolcic
- 1191 *Path MTU discovery* J. Mogul, S. Deering
- 1198 *FYI on the X window system* R. Scheifler
- 1207 *FYI on Questions and Answers: Answers to commonly asked "experienced Internet user" questions* G. Malkin, A. Marine, J. Reynolds
- 1208 *Glossary of networking terms* O. Jacobsen, D. 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. Rose
- 1227 *SNMP MUX protocol and MIB* M.T. Rose
- 1228 *SNMP-DPI: Simple Network Management Protocol Distributed Program Interface* G. 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
- 1256 *ICMP Router Discovery Messages* S. Deering, Ed.
- 1267 *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.
- 1285 *FDDI Management Information Base* J. Case
- 1315 *Management Information Base for Frame Relay DTEs* C. Brown, F. Baker, C. Carvalho
- 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. Malkin, A. Marine
- 1327 *Mapping between X.400 (1988)/ISO 10021 and RFC 822* S. Hardcastle-Kille
- 1340 *Assigned Numbers* J. Reynolds, J. Postel
- 1344 *Implications of MIME for Internet Mail Gateways* N. Bornstein
- 1349 *Type of Service in the Internet Protocol Suite* P. Almquist
- 1350 *The TFTP Protocol (Revision 2)* 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
- 1358 *Charter of the Internet Architecture Board (IAB)* L. Chapin
- 1363 *A Proposed Flow Specification* C. Partridge
- 1368 *Definition of Managed Objects for IEEE 802.3 Repeater Devices* D. McMaster, K. McCloghrie
- 1372 *Telnet Remote Flow Control Option* C. L. Hedrick, D. Borman
- 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 Extensions* G. Malkin, F. Baker
- 1390 *Transmission of IP and ARP over FDDI Networks* D. Katz
- 1393 *Traceroute Using an IP Option* G. Malkin
- 1398 *Definitions of Managed Objects for the Ethernet-Like Interface Types* F. Kastenholz
- 1408 *Telnet Environment Option* D. Borman, Ed.
- 1413 *Identification Protocol* M. St. Johns
- 1416 *Telnet Authentication Option* D. Borman, ed.
- 1420 *SNMP over IPX* S. Bostock
- 1428 *Transition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME* G. Vaudreuil
- 1442 *Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser
- 1443 *Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser
- 1445 *Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)* J. Galvin, K. McCloghrie
- 1447 *Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)* K. McCloghrie, J. Galvin
- 1448 *Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser
- 1464 *Using the Domain Name System to Store Arbitrary String Attributes* R. Rosenbaum
- 1469 *IP Multicast over Token-Ring Local Area Networks* T. Pusateri
- 1483 *Multiprotocol Encapsulation over ATM Adaptation Layer 5* Juha Heinanen
- 1497 *BOOTP Vendor Information Extensions* J. Reynolds
- 1514 *Host Resources MIB* P. Grillo, S. Waldbusser
- 1516 *Definitions of Managed Objects for IEEE 802.3 Repeater Devices* D. McMaster, K. McCloghrie

- 1521 *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies* N. Borenstein, N. Freed
- 1533 *DHCP Options and BOOTP Vendor Extensions* S. Alexander, R. Droms
- 1534 *Interoperation Between DHCP and BOOTP* R. Droms
- 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 *Internet Official Protocol Standards* J. Postel
- 1541 *Dynamic Host Configuration Protocol* R. Droms
- 1542 *Clarifications and Extensions for the Bootstrap Protocol* W. Wimer
- 1571 *Telnet Environment Option Interoperability Issues* D. Borman
- 1572 *Telnet Environment Option* S. Alexander
- 1573 *Evolution of the Interfaces Group of MIB-II* K. McCloghrie, F. Kastenholz
- 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. Marine, J. Reynolds, G. Malkin
- 1644 *T/TCP — TCP Extensions for Transactions Functional Specification* R. Braden
- 1646 *TN3270 Extensions for LUname and Printer Selection* C. Graves, T. Butts, M. Angel
- 1647 *TN3270 Enhancements* B. Kelly
- 1652 *SMTP Service Extension for 8bit-MIMEtransport* J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker
- 1664 *Using the Internet DNS to Distribute RFC1327 Mail Address Mapping Tables* C. Allochio, A. Bonito, B. Cole, S. Giordano, R. Hagens
- 1693 *An Extension to TCP: Partial Order Service* T. Connolly, P. Amer, P. Conrad
- 1695 *Definitions of Managed Objects for ATM Management Version 8.0 using SMIPv2* M. Ahmed, K. Tesink
- 1701 *Generic Routing Encapsulation (GRE)* S. Hanks, T. Li, D. Farinacci, P. Traina
- 1702 *Generic Routing Encapsulation over IPv4 networks* S. Hanks, T. Li, D. Farinacci, P. Traina
- 1706 *DNS NSAP Resource Records* B. Manning, R. Colella
- 1712 *DNS Encoding of Geographical Location* C. Farrell, M. Schulze, S. Pleitner D. Baldoni
- 1713 *Tools for DNS debugging* A. Romao

- 1723 *RIP Version 2—Carrying Additional Information* G. Malkin
- 1752 *The Recommendation for the IP Next Generation Protocol* S. Bradner, A. Mankin
- 1766 *Tags for the Identification of Languages* H. Alvestrand
- 1771 *A Border Gateway Protocol 4 (BGP-4)* Y. Rekhter, T. Li
- 1794 *DNS Support for Load Balancing* T. Brisco
- 1819 *Internet Stream Protocol Version 2 (ST2) Protocol Specification—Version ST2+* L. Delgrossi, L. Berger Eds.
- 1826 *IP Authentication Header* R. Atkinson
- 1828 *IP Authentication using Keyed MD5* P. Metzger, W. Simpson
- 1829 *The ESP DES-CBC Transform* P. Karn, P. Metzger, W. Simpson
- 1830 *SMTP Service Extensions for Transmission of Large and Binary MIME Messages* G. Vaudreuil
- 1832 *XDR: External Data Representation Standard* R. Srinivasan
- 1850 *OSPF Version 2 Management Information Base* F. Baker, R. Coltun
- 1854 *SMTP Service Extension for Command Pipelining* N. Freed
- 1869 *SMTP Service Extensions* J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker
- 1870 *SMTP Service Extension for Message Size Declaration* J. Klensin, N. Freed, K. Moore
- 1876 *A Means for Expressing Location Information in the Domain Name System* C. Davis, P. Vixie, T. Goodwin, I. Dickinson
- 1883 *Internet Protocol, Version 6 (IPv6) Specification* S. Deering, R. Hinden
- 1884 *IP Version 6 Addressing Architecture* R. Hinden, S. Deering, Eds.
- 1886 *DNS Extensions to support IP version 6* S. Thomson, C. Huitema
- 1888 *OSI NSAPs and IPv6* J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd
- 1891 *SMTP Service Extension for Delivery Status Notifications* K. Moore
- 1892 *The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages* G. Vaudreuil
- 1894 *An Extensible Message Format for Delivery Status Notifications* K. Moore, G. Vaudreuil
- 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 *Protocol 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
- 1930 *Guidelines for creation, selection, and registration of an Autonomous System (AS)* J. Hawkinson, T. Bates
- 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
- 1985 *SMTP Service Extension for Remote Message Queue Starting* J. De Winter
- 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 SMIPv2* K. McCloghrie, Ed.
- 2012 *SNMPv2 Management Information Base for the Transmission Control Protocol using SMIPv2* K. McCloghrie, Ed.
- 2013 *SNMPv2 Management Information Base for the User Datagram Protocol using SMIPv2* K. McCloghrie, Ed.
- 2018 *TCP Selective Acknowledgement Options* M. Mathis, J. Mahdavi, S. Floyd, A. Romanow
- 2026 *The Internet Standards Process — Revision 3* S. Bradner
- 2030 *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI* D. Mills
- 2033 *Local Mail Transfer Protocol* J. Myers
- 2034 *SMTP Service Extension for Returning Enhanced Error Codes* N. Freed
- 2040 *The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms* R. Baldwin, R. Rivest
- 2045 *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* N. Freed, N. Borenstein
- 2052 *A DNS RR for specifying the location of services (DNS SRV)* A. Gulbrandsen, P. Vixie
- 2065 *Domain Name System Security Extensions* D. Eastlake 3rd, C. Kaufman
- 2066 *TELNET CHARSET Option* R. Gellens
- 2080 *RIPng for IPv6* G. Malkin, R. Minnear

- 2096 *IP Forwarding Table MIB* F. Baker
- 2104 *HMAC: Keyed-Hashing for Message Authentication* H. Krawczyk, M. Bellare, R. Canetti
- 2119 *Keywords for use in RFCs to Indicate Requirement Levels* S. Bradner
- 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
- 2136 *Dynamic Updates in the Domain Name System (DNS UPDATE)* P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound
- 2137 *Secure Domain Name System Dynamic Update* D. Eastlake 3rd
- 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 Functional Specification* R. Braden, Ed., 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
- 2217 *Telnet Com Port Control Option* G. Clarke
- 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
- 2251 *Lightweight Directory Access Protocol (v3)* M. Wahl, T. Howes, S. Kille
- 2253 *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names* M. Wahl, S. Kille, T. Howes
- 2254 *The String Representation of LDAP Search Filters* T. Howes
- 2261 *An Architecture for Describing SNMP Management Frameworks* D. Harrington, R. Presuhn, B. Wijnen
- 2262 *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* J. Case, D. Harrington, R. Presuhn, B. Wijnen
- 2271 *An Architecture for Describing SNMP Management Frameworks* D. Harrington, R. Presuhn, B. Wijnen

- 2273 *SNMPv3 Applications* D. Levi, P. Meyer, B. Stewartz
- 2274 *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* U. Blumenthal, B. Wijnen
- 2275 *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* B. Wijnen, R. Presuhn, K. McCloghrie
- 2292 *Advanced Sockets API for IPv6* W. Stevens, M. Thomas
- 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 (IPOA-MIB)* 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
- 2358 *Definitions of Managed Objects for the Ethernet-like Interface Types* J. Flick, J. Johnson
- 2373 *IP Version 6 Addressing Architecture* R. Hinden, 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
- 2385 *Protection of BGP Sessions via the TCP MD5 Signature Option* A. Hefferman
- 2389 *Feature negotiation mechanism for the File Transfer Protocol* P. Hethmon, R. Elz
- 2401 *Security Architecture for Internet Protocol* S. Kent, R. Atkinson
- 2402 *IP Authentication Header* S. Kent, R. Atkinson
- 2403 *The Use of HMAC-MD5-96 within ESP and AH* C. Madson, R. Glenn
- 2404 *The Use of HMAC-SHA-1-96 within ESP and AH* C. Madson, R. Glenn
- 2405 *The ESP DES-CBC Cipher Algorithm With Explicit IV* C. Madson, N. Doraswamy
- 2406 *IP Encapsulating Security Payload (ESP)* S. Kent, R. Atkinson
- 2407 *The Internet IP Security Domain of Interpretation for ISAKMPD* Piper
- 2408 *Internet Security Association and Key Management Protocol (ISAKMP)* D. Maughan, M. Schertler, M. Schneider, J. Turner
- 2409 *The Internet Key Exchange (IKE)* D. Harkins, D. Carrel
- 2410 *The NULL Encryption Algorithm and Its Use With IPsec* R. Glenn, S. Kent,
- 2428 *FTP Extensions for IPv6 and NATs* M. Allman, S. Ostermann, C. Metz
- 2445 *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* F. Dawson, D. Stenerson
- 2459 *Internet X.509 Public Key Infrastructure Certificate and CRL Profile* R. Housley, W. Ford, W. Polk, D. Solo
- 2460 *Internet Protocol, Version 6 (IPv6) Specification* 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
- 2463 *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification* A. Conta, S. Deering
- 2464 *Transmission of IPv6 Packets over Ethernet Networks* M. Crawford
- 2466 *Management Information Base for IP Version 6: ICMPv6 Group* D. Haskin, S. Onishi
- 2476 *Message Submission* R. Gellens, J. Klensin
- 2487 *SMTP Service Extension for Secure SMTP over TLS* P. Hoffman
- 2505 *Anti-Spam Recommendations for SMTP MTAs* G. Lindberg
- 2523 *Photuris: Extended Schemes and Attributes* P. Karn, W. Simpson
- 2535 *Domain Name System Security Extensions* D. Eastlake 3rd
- 2538 *Storing Certificates in the Domain Name System (DNS)* D. Eastlake 3rd, O. Gudmundsson
- 2539 *Storage of Diffie-Hellman Keys in the Domain Name System (DNS)* D. Eastlake 3rd
- 2540 *Detached Domain Name System (DNS) Information* D. Eastlake 3rd
- 2554 *SMTP Service Extension for Authentication* J. Myers
- 2570 *Introduction to Version 3 of the Internet-standard Network Management Framework* J. Case, R. Mundy, D. Partain, B. Stewart
- 2571 *An Architecture for Describing SNMP Management Frameworks* B. Wijnen, D. Harrington, R. Presuhn
- 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
- 2576 *Co-Existence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework* R. Frye, D. Levi, S. Routhier, B. Wijnen
- 2578 *Structure of Management Information Version 2 (SMIv2)* K. McCloghrie, D. Perkins, J. Schoenwaelder
- 2579 *Textual Conventions for SMIv2* K. McCloghrie, D. Perkins, J. Schoenwaelder
- 2580 *Conformance Statements for SMIv2* K. McCloghrie, D. Perkins, J. Schoenwaelder
- 2581 *TCP Congestion Control* M. Allman, V. Paxson, W. Stevens
- 2583 *Guidelines for Next Hop Client (NHC) Developers* R. Carlson, L. Winkler
- 2591 *Definitions of Managed Objects for Scheduling Management Operations* D. Levi, J. Schoenwaelder
- 2625 *IP and ARP over Fibre Channel* M. Rajagopal, R. Bhagwat, W. Rickard

- 2635 *Don't SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*)* S. Hambridge, A. Lunde
- 2637 *Point-to-Point Tunneling Protocol* K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, G. Zorn
- 2640 *Internationalization of the File Transfer Protocol* B. Curtin
- 2665 *Definitions of Managed Objects for the Ethernet-like Interface Types* J. Flick, J. Johnson
- 2671 *Extension Mechanisms for DNS (EDNS0)* P. Vixie
- 2672 *Non-Terminal DNS Name Redirection* M. Crawford
- 2675 *IPv6 Jumbograms* D. Borman, S. Deering, R. Hinden
- 2710 *Multicast Listener Discovery (MLD) for IPv6* S. Deering, W. Fenner, B. Haberman
- 2711 *IPv6 Router Alert Option* C. Partridge, A. Jackson
- 2740 *OSPF for IPv6* R. Coltun, D. Ferguson, J. Moy
- 2753 *A Framework for Policy-based Admission Control* R. Yavatkar, D. Pendarakis, R. Guerin
- 2758 *Definitions of Managed Objects for Service Level Agreements Performance Monitoring* K. White
- 2782 *A DNS RR for specifying the location of services (DNS SRV)* A. Gubrandsen, P. Vixie, L. Esibov
- 2821 *Simple Mail Transfer Protocol* J. Klensin, Ed.
- 2822 *Internet Message Format* P. Resnick, Ed.
- 2840 *TELNET KERMIT OPTION* J. Altman, F. da Cruz
- 2845 *Secret Key Transaction Authentication for DNS (TSIG)* P. Vixie, O. Gudmundsson, D. Eastlake 3rd, B. Wellington
- 2851 *Textual Conventions for Internet Network Addresses* M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder
- 2852 *Deliver By SMTP Service Extension* D. Newman
- 2874 *DNS Extensions to Support IPv6 Address Aggregation and Renumbering* M. Crawford, C. Huitema
- 2915 *The Naming Authority Pointer (NAPTR) DNS Resource Record* M. Mealling, R. Daniel
- 2920 *SMTP Service Extension for Command Pipelining* N. Freed
- 2930 *Secret Key Establishment for DNS (TKEY RR)* D. Eastlake, 3rd
- 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
- 2992 *Analysis of an Equal-Cost Multi-Path Algorithm* C. Hopps

- 3019 *IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol* B. Haberman, R. Worzella
- 3060 *Policy Core Information Model—Version 1 Specification* B. Moore, E. Ellessen, J. Strassner, A. Westerinen
- 3152 *Delegation of IP6.ARPA* R. Bush
- 3291 *Textual Conventions for Internet Network Addresses* M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder
- 3363 *Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System* R. Bush, A. Durand, B. Fink, O. Gudmundsson, T. Hain
- 3390 *Increasing TCP's Initial Window* M. Allman, S. Floyd, C. Partridge
- 3411 *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks* D. Harrington, R. Presuhn, B. Wijnen
- 3412 *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* J. Case, D. Harrington, R. Presuhn, B. Wijnen
- 3413 *Simple Network Management Protocol (SNMP) Applications* D. Levi, P. Meyer, B. Stewart
- 3414 *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* U. Blumenthal, B. Wijnen
- 3415 *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* B. Wijnen, R. Presuhn, K. McCloghrie
- 3419 *Textual Conventions for Transport Addresses* M. Daniele, J. Schoenwaelder
- 3484 *Default Address Selection for Internet Protocol version 6 (IPv6)* R. Draves
- 3493 *Basic Socket Interface Extensions for IPv6* R. Gilligan, S. Thomson, J. Bound, J. McCann, W. Stevens
- 3513 *Internet Protocol Version 6 (IPv6) Addressing Architecture* R. Hinden, S. Deering
- 3542 *Advanced Sockets Application Programming Interface (API) for IPv6* W. Richard Stevens, M. Thomas, E. Nordmark, T. Jinmei
- 3658 *Delegation Signer (DS) Resource Record (RR)* O. Gudmundsson
- 3715 *IPsec-Network Address Translation (NAT) Compatibility Requirements* B. Aboba, W. Dixon
- 3947 *Negotiation of NAT-Traversal in the IKE* T. Kivinen, B. Swander, A. Huttunen, V. Volpe
- 3948 *UDP Encapsulation of IPsec ESP Packets* A. Huttunen, B. Swander, V. Volpe, L. DiBurro, M. Stenberg

Internet drafts

Internet drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Other groups may also distribute working documents as Internet drafts. You can see Internet drafts at <http://www.ietf.org/ID.html>.

Several areas of IPv6 implementation include elements of the following Internet drafts and are subject to change during the RFC review process.

**Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6
(IPv6) Specification**
A. Conta, S. Deering

Appendix C. 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 V1R7 are complete except for the updates contained in the information APARs that might be issued after V1R7 documents went to press.
2. Information APARs are predefined for z/OS V1R7 Communications Server and might not contain updates.
3. 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. For information APARs for V1R7, see <http://www.ibm.com/support/docview.wss?uid=swg21178966>.

Table 2. IP information APARs for z/OS Communications Server

Title	V1R6	V1R5	V1R4
New Function Summary (both IP and SNA)	II13824		
Quick Reference (both IP and SNA)	II13831		II13246
IP and SNA Codes	II13842		II13254
IP API Guide	II13844	II13577	II13255 II13790
IP CICS Sockets Guide		II13578	II13257
IP Configuration Guide	II13826	II13568	II13244 II13541 II13652 II13646
IP Configuration Reference	II13827	II13569 II13789	II13245 II13521 II13647 II13739
IP Diagnosis	II13836	II13571	II13249 II13493
IP Messages Volume 1	II13838	II13572	II13624 II13250
IP Messages Volume 2	II13839	II13573	II13251
IP Messages Volume 3	II13840	II13574	II13252
IP Messages Volume 4	II13841	II13575	II13253 II13628
IP Migration		II13566	II13242 II13738
IP Network and Application Design Guide	II13825	II13567	II13243

Table 2. IP information APARs for z/OS Communications Server (continued)

Title	V1R6	V1R5	V1R4
IP Network Print Facility			
IP Programmer's Reference	II13843	II13581	II13256
IP User's Guide and Commands	II13832	II13570	II13247
IP System Admin Commands	II13833	II13580	II13248 II13792

Information APARs for SNA documents

Table 3 lists information APARs for SNA documents. For information APARs for V1R7, see <http://www.ibm.com/support/docview.wss?uid=swg21178966>.

Table 3. SNA information APARs for z/OS Communications Server

Title	V1R6	V1R5	V1R4
New Function Summary (both IP and SNA)	II13824		
Quick Reference (both IP and SNA)	II13831		II13246
IP and SNA Codes	II13842		II13254
SNA Customization	II13857	II13560	II13240
SNA Diagnosis		II13558	II13236 II13735
SNA Diagnosis, Vol. 1: Techniques and Procedures	II13852		
SNA Diagnosis, Vol. 2: FFST Dumps and the VIT	II13853		
SNA Messages	II13854	II13559	II13238 II13736
SNA Network Implementation Guide	II13849	II13555	II13234 II13733
SNA Operation	II13851	II13557	II13237
SNA Migration		II13554	II13233 II13732
SNA Programming	II13858		II13241
SNA Resource Definition Reference	II13850	II13556	II13235 II13734
SNA Data Areas, Vol. 1 and 2			II13239
SNA Data Areas, 1	II13855		
SNA Data Areas, 2	II13856		

Other information APARs

Table 4 lists information APARs not related to documents.

Table 4. Non-document information APARs

Content	Number
Index to APARs that list recommended VTAM maintenance	II11220

Table 4. Non-document information APARs (continued)

	Content	Number
I	Index to APARs that list trace and dump requests for VTAM problems	II13202
	Index of Communication Server IP information APARs	II12028
I	MPC and CTC	II01501
	Collecting TCPIP CTRACES	II12014
I	CSM for VTAM	II13442
I	CSM for TCP/IP	II13951
I	DLUR/DLUS for z/OS V1R2, V1R4, and V1R5	II12986, II13456, and II13783
	DOCUMENTATION REQUIRED FOR OSA/2, OSA EXPRESS AND OSA QDIO	II13016
	DYNAMIC VIPA (BIND)	II13215
	DNS — common problems and solutions	II13453
	Enterprise Extender	II12223
	FTPing doc to z/OS Support	II12030
	FTP problems	II12079
	Generic resources	II10986
	HPR	II10953
I	iQDIO	II13142
	LPR problems	II12022
	MNPS	II10370
	NCPROUTE problems	II12025
	OMPROUTE	II12026
	PASCAL API	II11814
	Performance	II11710 II11711 II11712
	Resolver	II13398 II13399 II13452
	Socket API	II11996 II12020
	SMTP problems	II12023
	SNMP	II13477 II13478
	SYSLOGD howto	II12021
	TCPIP connection states	II12449
	Telnet	II11574 II13135
	TN3270 TELNET SSL common problems	II13369

Appendix D. 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 such products 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 Vol 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.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:
www.ibm.com/servers/eserver/zseries/zos/bkserv/

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.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California. Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Portions of this publication relating to RPC are Copyright © Sun Microsystems, Inc., 1988, 1989.

Some portions of this publication relating to X Window System** are Copyright © 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, and the Massachusetts Institute Of Technology, Cambridge, Massachusetts. All Rights Reserved.

Some portions of this publication relating to X Window System are Copyright © 1986, 1987, 1988 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute the M.I.T., Digital Equipment Corporation, and Hewlett-Packard Corporation portions of this software and its documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T., Digital, and Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T., Digital, and Hewlett-Packard make no representation about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright © 1983, 1995-1997 Eric P. Allman

Copyright © 1988, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software program contains code, and/or derivatives or modifications of code originating from the software program "Popper." Popper is Copyright ©1989-1991 The Regents of the University of California, All Rights Reserved. Popper was created by Austin Shelton, Information Systems and Technology, University of California, Berkeley.

Permission from the Regents of the University of California to use, copy, modify, and distribute the "Popper" software contained herein for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies. HOWEVER, ADDITIONAL PERMISSIONS MAY BE NECESSARY FROM OTHER PERSONS OR ENTITIES, TO USE DERIVATIVES OR MODIFICATIONS OF POPPER.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THE POPPER SOFTWARE, OR ITS DERIVATIVES OR MODIFICATIONS, AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE POPPER SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Copyright © 1983 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1991, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 1990 by the Massachusetts Institute of Technology

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore

if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original M.I.T. software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright © 1998 by the FundsXpress, INC. All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be

given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)". The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related.
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include acknowledgement:
"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

This product includes cryptographic software written by Eric Young.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 2004 IBM Corporation and its licensors, including Sendmail, Inc., and the Regents of the University of California. All rights reserved.

Copyright © 1999,2000,2001 Compaq Computer Corporation

Copyright © 1999,2000,2001 Hewlett-Packard Company

Copyright © 1999,2000,2001 IBM Corporation

Copyright © 1999,2000,2001 Hummingbird Communications Ltd.

Copyright © 1999,2000,2001 Silicon Graphics, Inc.

Copyright © 1999,2000,2001 Sun Microsystems, Inc.

Copyright © 1999,2000,2001 The Open Group

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

X Window System is a trademark of The Open Group.

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:

Advanced Peer-to-Peer Networking	MVS/SP
AFP	MVS/XA
AD/Cycle	NetView
AIX	Network Station
AIX/ESA	Nways
AnyNet	Notes
APL2	OfficeVision/MVS
AS/400	OfficeVision/VM
AT	Open Class
BookManager	OS/2
BookMaster	OS/390
C/370	OS/400
CICS	Parallel Sysplex
CICS/ESA	PR/SM
C/MVS	PROFS
Common User Access	PS/2
C Set ++	RACF
CT	Redbooks
CUA	Resource Link
DB2	RETAIN
DFSMSdfp	RISC System/6000
DFSMSshsm	RMF
DFSMS/MVS	RS/6000
DPI	S/370
Domino	S/390
DRDA	S/390 Parallel Enterprise Server
Enterprise Systems Architecture/370	SAA
ESCON	SecureWay
eServer	SP
ES/3090	SP2
ES/9000	SQL/DS
ES/9370	System/360
EtherStreamer	System/370
Extended Services	System/390
FFST	SystemView
FFST/2	Tivoli
First Failure Support Technology	TURBOWAYS
GDDM	VM/ESA
IBM	VSE/ESA
IBMLink	VTAM
IMS	WebSphere
IMS/ESA	XT
HiperSockets	z/Architecture
Language Environment	z/OS
LANStreamer	zSeries
Library Reader	z/VM
LPDA	400
Micro Channel	3090
Multiprise	3890
MVS	
MVS/DFP	
MVS/ESA	

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.

| Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the
| United States, other countries, or both.

| Linux is a trademark of Linus Torvalds in the United States, other countries, or
| both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

| Intel is a registered trademark of Intel Corporation or its subsidiaries in the United
| States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

z/OS Communications Server information

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

z/OS Communications Server documentation is available:

- Online at the z/OS Internet Library web page at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv>
- In softcopy on CD-ROM collections. See “Softcopy information” on page xi.

z/OS Communications Server library

z/OS 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 C, “Information APARs,” on page 571 for a list of the documents and the info APARs associated with them.

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

Title	Number	Description
<i>z/OS Communications Server: New Function Summary</i>	GC31-8771	This document is intended to help you plan for new IP for SNA function, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions.
<i>z/OS Communications Server: IPv6 Network and Application Design Guide</i>	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
<i>z/OS Communications Server: IP Configuration Guide</i>	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 <i>z/OS Communications Server: IP Configuration Reference</i> .

Title	Number	Description
<i>z/OS Communications Server: IP Configuration Reference</i>	SC31-8776	This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the <i>z/OS Communications Server: IP Configuration Guide</i> . The information in this document includes: <ul style="list-style-type: none"> • TCP/IP configuration data sets • Configuration statements • Translation tables • SMF records • Protocol number and port assignments
<i>z/OS Communications Server: SNA Network Implementation Guide</i>	SC31-8777	This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the <i>z/OS Communications Server: SNA Resource Definition Reference</i> .
<i>z/OS Communications Server: SNA Resource Definition Reference</i>	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 <i>z/OS Communications Server: SNA Network Implementation Guide</i> .
<i>z/OS Communications Server: SNA Resource Definition Samples</i>	SC31-8836	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
<i>z/OS Communications Server: AnyNet SNA over TCP/IP</i>	SC31-8832	This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.
<i>z/OS Communications Server: AnyNet Sockets over SNA</i>	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.
<i>z/OS Communications Server: IP Network Print Facility</i>	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
<i>z/OS Communications Server: IP User's Guide and Commands</i>	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.
<i>z/OS Communications Server: IP System Administrator's Commands</i>	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.
<i>z/OS Communications Server: SNA Operation</i>	SC31-8779	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
<i>z/OS Communications Server: Quick Reference</i>	SX75-0124	This document contains essential information about SNA and IP commands.

Customization

Title	Number	Description
<i>z/OS Communications Server: SNA Customization</i>	SC31-6854	<p>This document enables you to customize SNA, and includes the following:</p> <ul style="list-style-type: none"> • 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
<i>z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference</i>	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.
<i>z/OS Communications Server: IP CICS Sockets Guide</i>	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.
<i>z/OS Communications Server: IP IMS Sockets Guide</i>	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.
<i>z/OS Communications Server: IP Programmer's Guide and Reference</i>	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.
<i>z/OS Communications Server: SNA Programming</i>	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.
<i>z/OS Communications Server: SNA Programmer's LU 6.2 Guide</i>	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.)
<i>z/OS Communications Server: SNA Programmer's LU 6.2 Reference</i>	SC31-8810	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.
<i>z/OS Communications Server: CSM Guide</i>	SC31-8808	This document describes how applications use the communications storage manager.

Title	Number	Description
<i>z/OS Communications Server: CMIP Services and Topology Agent Guide</i>	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
<i>z/OS Communications Server: IP Diagnosis Guide</i>	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.
<i>z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures</i> and <i>z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT</i>	GC31-6850 GC31-6851	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.
<i>z/OS Communications Server: SNA Data Areas Volume 1</i> and <i>z/OS Communications Server: SNA Data Areas Volume 2</i>	GC31-6852 GC31-6853	These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and codes

Title	Number	Description
<i>z/OS Communications Server: SNA Messages</i>	SC31-8790	This document describes the ELM, IKT, IST, ISU, IUT, IVT, and USS messages. Other information in this document includes: <ul style="list-style-type: none"> • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
<i>z/OS Communications Server: IP Messages Volume 1 (EZA)</i>	SC31-8783	This volume contains TCP/IP messages beginning with EZA.
<i>z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)</i>	SC31-8784	This volume contains TCP/IP messages beginning with EZB or EZD.
<i>z/OS Communications Server: IP Messages Volume 3 (EZY)</i>	SC31-8785	This volume contains TCP/IP messages beginning with EZY.
<i>z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)</i>	SC31-8786	This volume contains TCP/IP messages beginning with EZZ and SNM.
<i>z/OS Communications Server: IP and SNA Codes</i>	SC31-8791	This document describes codes and other information that appear in z/OS Communications Server messages.

APPC Application Suite

Title	Number	Description
<i>z/OS Communications Server: APPC Application Suite User's Guide</i>	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.

Title	Number	Description
<i>z/OS Communications Server: APPC Application Suite Administration</i>	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.
<i>z/OS Communications Server: APPC Application Suite Programming</i>	SC31-8834	This document provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

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.

Please send your comments to us in either of the following ways:

- If you prefer to send comments by FAX, use this number: 1+919-254-4028
- If you prefer to send comments electronically, use this address:
 - comsvrcf@us.ibm.com.
- If you prefer to send comments by post, use this address:
 - International Business Machines Corporation
 - Attn: z/OS Communications Server Information Development
 - P.O. Box 12195, 3039 Cornwallis Road
 - Department AKCA, Building 501
 - Research Triangle Park, North Carolina 27709-2195

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.



Program Number: 5694-A01 and 5655-G52

Printed in USA

SC31-8785-05



Spine information:



z/OS Communications Server

z/OS V1R7.0 Comm Svr: IP Messages Volume 3 (EZY)

Version 1
Release 7