z/OS Communications Server

IP Messages: Volume 4 (EZZ-SNM)

Version 1 Release 4

SC31-8786-02

z/OS Communications Server

IP Messages: Volume 4 (EZZ-SNM)

Version 1 Release 4

SC31-8786-02

Note:
Before using this information and the product it supports, be sure to read the general information under “Notices” on page 749.

Third Edition (September 2002)

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

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

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

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

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

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

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

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

IBMLink™:
CIBMORCF at RALVM17

IBM Mail Exchange:
tkinlaw@us.ibm.com

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

© Copyright International Business Machines Corporation 1994, 2002. 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 use this document v
Terms and abbreviations used in this document v
How the term “internet” is used in this document v
How the document distinguishes “TCPIP” from “TCP/IP” v
Other abbreviations and terms. v
The IBM TCP/IP message standards vi
Older numbering convention vi
Current numbering convention viii
Message ranges by volume ix
Where to find more information ix
Where to find related information on the Internet ix
Accessing z/OS licensed documents on the Internet. x
Using LookAt to look up message explanations x
How to contact IBM service xi
z/OS Communications Server information xi

Summary of changes xix

Chapter 1. EZZ0*xxxx* messages 1

Chapter 2. EZZ2*xxxx* messages 61

Chapter 3. EZZ3*xxxx* messages 71

Chapter 4. EZZ4*xxxx* messages 99

Chapter 5. EZZ5*xxxx* messages. 155

Chapter 6. EZZ6*xxxx* messages. 163

Chapter 7. EZZ7*xxxx* messages. 321

Chapter 8. EZZ8*xxxx* messages. 427

Chapter 9. EZZ9*xxxx* messages. 551

Chapter 10. SNM*xxxxx* messages 713

Appendix A. Message ranges and the components that issue them . . . 729
Volume 1 ranges. 729
Volume 2 ranges. 729
Volume 3 ranges. 729
Volume 4 ranges. 730

Appendix B. Related protocol specifications (RFCs) 733
Draft RFCs 740

Appendix C. Information APARs 743
Information APARs for IP documents 743
Information APARs for SNA documents 744
Other information APARs. 744

© Copyright IBM Corp. 1994, 2002

Appendix D. Accessibility 747
Using assistive technologies 747
Keyboard navigation of the user interface. 747

Notices 749
Trademarks. 752

Communicating Your Comments to IBM 755

About this document

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

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

This document supports z/OS.e™ .

Who should use this document

This document assists TCP/IP operators, system programmers, and users to:
- Analyze a problem
- Classify the problem as a specific type
- Describe the problem to the IBM Software Support Center

Familiarity with TCP/IP concepts and terms is assumed.

Terms and abbreviations used in this document

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

How the term "internet" is used in this document

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

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

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

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

Other abbreviations and terms

The following abbreviations are also used in this document:

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

RPC Sun Microsystems' implementation of remote procedure calls.

© Copyright IBM Corp. 1994, 2002

SQL IBM Structured Query Language.

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

ES/9000®
Enterprise System/9000 processor.

ES/9370™
Enterprise System/9370 processor.

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

Term Description

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

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

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

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

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

The IBM TCP/IP message standards

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

Older numbering convention

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

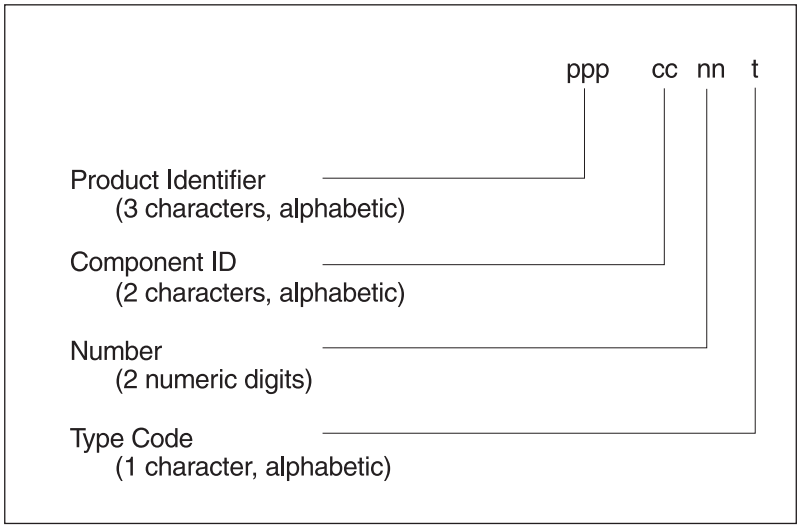

Figure 1. Sample message formats

The various type code characters are defined below.

Type Code
Severity

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

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

I Informational message.

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

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

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

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

Table 1. EZY message component identifiers

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

Table 1. EZY message component identifiers (continued)

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

Current numbering convention

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

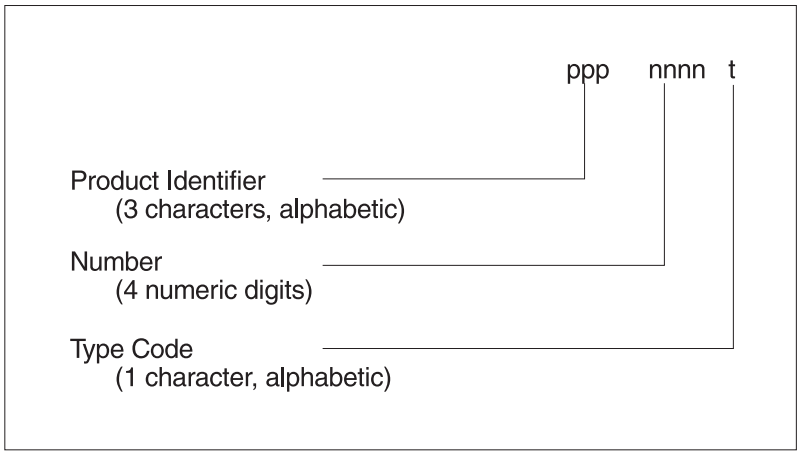

Figure 2. Sample message formats

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

For example:

EZZ0902I Use a host name after HOST option.

The various type code characters are defined below.

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

Message ranges by volume

The IP Messages are distributed among volumes as follows:

Volume 1
EZA Messages

Volume 2
EZB Messages

Volume 3
EZY Messages

Volume 4
EZZ and SNM Messages

Where to find more information

This section contains:

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

Where to find related information on the Internet

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

z/OS Internet Library
– http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

IBM Communications Server product
– http://www.software.ibm.com/network/commserver/

IBM Communications Server product support
– http://www.software.ibm.com/network/commserver/support/

IBM Systems Center publications
– http://www.redbooks.ibm.com/

IBM Systems Center flashes
– http://www-1.ibm.com/support/techdocs/atsmastr.nsf

RFCs
– http://www.ietf.org/rfc.html

RFC drafts
– http://www.ietf.org/ID.html

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

DNS web sites

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

USENET news groups:
comp.protocols.dns.bind

For BIND mailing lists, see:

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

- BIND Users
  - Subscribe by sending mail to bind-users-request@isc.org.
  - Submit questions or answers to this forum by sending mail to bind-users@isc.org.
- BIND 9 Users (Note: This list may not be maintained indefinitely.)
  - Subscribe by sending mail to bind9-users-request@isc.org.
  - Submit questions or answers to this forum by sending mail to bind9-users@isc.org.

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

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

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

Accessing z/OS licensed documents on the Internet

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

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

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

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

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

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

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

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

Printed licensed documents are not available from IBM.

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

Using LookAt to look up message explanations

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

You can access LookAt from the Internet at:

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

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

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

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

How to contact IBM service

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

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

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

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

If you would like to provide feedback on this publication, see “Communicating Your Comments to IBM” on page 755.

z/OS Communications Server information

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

z/OS Communications Server publications are available:

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

Softcopy information

Softcopy publications are available in the following collections:

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

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

z/OS Communications Server library

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

Updates to documents are available on RETAIN® and in information APARs (info APARs). See Appendix C, “Information APARs” on page 743 for a list of the documents and the info APARs associated with them.

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

Planning and migration:

Title	Number	Description
z/OS Communications Server: SNA Migration	GC31-8774	This document is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This document also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.
z/OS Communications Server: IP Migration	GC31-8773	This document is intended to help you plan for TCP/IP Services, whether you are migrating from a previous version or installing IP for the first time. This document also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with TCP/IP Services.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC31-8885	This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server’s support of IPv6, coexistence with IPv4, and migration issues.

Resource definition, configuration, and tuning:

Title	Number	Description
z/OS Communications Server: IP Configuration Guide	SC31-8775	This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX® System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Reference*.
z/OS Communications Server: IP Configuration Reference	SC31-8776	This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Guide*. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • SMF records • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC31-8777	This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the *z/OS Communications Server: SNA Resource Definition Reference*.
z/OS Communications Server: SNA Resource Definition Reference	SC31-8778	This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document in conjunction with the *z/OS Communications Server: SNA Network Implementation Guide*.
z/OS Communications Server: SNA Resource Definition Samples	SC31-8836	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: AnyNet SNA over TCP/IP	SC31-8832	This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.
z/OS Communications Server: AnyNet Sockets over SNA	SC31-8831	This guide provides information to help you install, configure, use, and diagnose sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.
z/OS Communications Server: IP Network Print Facility	SC31-8833	This document is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Operation:

Title	Number	Description
z/OS Communications Server: IP User's Guide and Commands	SC31-8780	This document describes how to use TCP/IP applications. It contains requests that allow a user to log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC31-8781	This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.

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

Customization:

Title	Number	Description
z/OS Communications Server: SNA Customization	LY43-0092	This document enables you to customize SNA, and includes the following: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines

Writing application programs:

Title	Number	Description
z/OS Communications Server: IP Application Programming Interface Guide	SC31-8788	This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC31-8807	This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS® using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC31-8830	This document is for programmers who want application programs that use the IMS™ TCP/IP application development services provided by IBM's TCP/IP Services.
z/OS Communications Server: IP Programmer's Reference	SC31-8787	This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC31-8829	This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC31-8811	This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)

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

Diagnosis:

Title	Number	Description
z/OS Communications Server: IP Diagnosis	GC31-8782	This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures and *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT*	LY43-0088 LY43-0089	These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
z/OS Communications Server: SNA Data Areas Volume 1 and *z/OS Communications Server: SNA Data Areas Volume 2*	LY43-0090 LY43-0091	These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and codes:

Title	Number	Description
z/OS Communications Server: SNA Messages	SC31-8790	This document describes the ELM, IKT, IST, ISU, IUT, IVT, and USS messages. Other information in this document includes: • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
z/OS Communications Server: IP Messages Volume 1 (EZA)	SC31-8783	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB)	SC31-8784	This volume contains TCP/IP messages beginning with EZB.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC31-8785	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ-SNM)	SC31-8786	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC31-8791	This document describes codes and other information that appear in z/OS Communications Server messages.

APPC Application Suite:

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

Redbooks

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

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

Related information

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

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

The table below lists documents that may be helpful to readers.

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

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

Determining if a publication is current

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

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

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

Summary of changes

Summary of changes
for SC31-8786-02
z/OS Version 1 Release 4

This document contains information previously presented in SC31-8786-01, which supports z/OS Version 1 Release 2. The information in this document supports both IPv6 and IPv4. Unless explicitly noted, information describes IPv4 networking protocol. IPv6 support is qualified within the text.

New information

- EZZ0684I
- EZZ0695I–EZZ0697I
- EZZ0699I–EZZ0718I
- EZZ2364I
- EZZ2383I
- EZZ2389I
- EZZ2500I
- EZZ3129I–EZZ3133I
- EZZ4336I–EZZ4348I
- EZZ6007I
- EZZ6352I and EZZ6353I
- EZZ6355I and EZZ6356I
- EZZ6367I
- EZZ6374I and EZZ6375I
- EZZ9304I
- EZZ9550I–EZZ9554I
- EZZ9556I–EZZ9594I
- EZZ9599I
- EZZ9680I–EZZ9724I
- EZZ9780I–EZZ9784I
- EZZ9790I–EZZ9799I
- EZZ9809I–EZZ9811I

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

Deleted information

- EZZ2368I
- EZZ3100I
- EZZ3102I–EZZ3104I
- EZZ3124I
- EZZ6068I
- EZZ6360I
- EZZ6383I
- EZZ7251–EZZ7261
- EZZ7476I

© Copyright IBM Corp. 1994, 2002

Changed information

- EZZ0352I
- EZZ0657I
- EZZ2357I
- EZZ2369I
- EZZ2374I
- EZZ3105I–EZZ3133I — Minor adjustments to accomodate terminology and technical changes to the TSO and z/OS PING function.
- EZZ6351I–EZZ6388I — Minor adjustments to accomodate terminology and technical changes to the TSO and z/OS UNIX Traceroute functions.
- EZZ8821I–EZZ8823I
- EZZ9106I–EZZ9108I
- EZZ9110I–EZZ9112I
- EZZ9136I–EZZ9137I
- EZZ9140I–EZZ9145I
- EZZ9147I–EZZ9153I
- EZZ9188I
- EZZ9376I
- EZZ9535I
- EZZ9541I

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

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

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

This document supports z/OS.e.

Summary of changes
for SC31-8786-01
z/OS Version 1 Release 2

This document contains information previously presented in SC31-8786-00, which supports z/OS Version 1 Release 1.

New information

- EZZ0672 – EZZ0679
- EZZ0683
- EZZ0685 – EZZ0692
- EZZ2385 – EZZ2388
- EZZ3105
- EZZ4331 – EZZ4335

- EZZ4360 – EZZ4367
- EZZ5023 – EZZ5025
- EZZ6316
- EZZ6388
- EZZ7475 and EZZ7476
- EZZ8026
- EZZ8077
- EZZ8441 – EZZ8451
- EZZ8453 – EZZ8455
- EZZ8475 – EZZ8477
- EZZ8638 – EZZ8653
- EZZ8761 – EZZ8769
- EZZ8791 – EZZ9293
- EZZ9315 – EZZ9324
- EZZ9331 – EZZ9596

Deleted information

- EZZ0175 – EZZ0208
- EZZ6004
- EZZ6016
- EZZ6019
- EZZ6029 – EZZ6033
- EZZ6036
- EZZ6062
- EZZ6063
- EZZ6066
- EZZ6069 – EZZ6079
- EZZ8483 – EZZ8494
- EZZ8497
- EZZ8499

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

Summary of changes
for SC31-8786-00
z/OS Version 1 Release 1

This document contains information also presented in *OS/390 V2R10 IBM Communications Server: IP Messages Volume 3 (EZY-EZZ-SNM)*.

Chapter 1. EZZ0*xxxx* messages

EZZ0053I Command *cmd* completed successfully

Explanation: The command finished without error.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFC00, EZACFYAC

Procedure Name: parseCmd

EZZ0054I Error: Command *cmd* did not complete successfully

Explanation: The processing of the VARY TCPIP command did not complete successfully.

System Action: TCP/IP continues.

User or Operator Response: Check the system log or the SYSERROR DD file specified in your TCPIP procedure for possible reasons.

System Programmer Response: None.

Source Data Set: EZACFC00, EZACDSTR

Procedure Name: parseCmd

EZZ0059I *cmd* command failed: *reason*

Explanation: The command failed for the reason indicated

If the command you issued was VARY TCPIP,,OBEYFILE, the following are explanations for the possible *reasons*:
- NOT AUTHORIZED - your user ID does not have proper authorization to the MVS.VARY.TCPIP.OBEYFILE RACF® resource.
- SEE PREVIOUS MESSAGES - there were error messages generated while processing your profile.

If the command you issued was VARY DROP, the following are reasons for the error:
- INCORRECT CONNECTION NUMBER *number* - the connection number specified cannot be dropped.
- INTERNAL COMMAND *number* FAILED ERRNO=*errno* ERRNO2=*errno2* - an internal command was issued to TCP/IP.

System Action: TCP/IP continues.

User or Operator Response:

For the VARY TCPIP,,OBEYFILE command, the following are user responses for the possible *reasons*:
- NOT AUTHORIZED - make sure that your user ID has the MVS.VARY.TCPIP.OBEYFILE defined in the RACF profile and resubmit the command.
- SEE PREVIOUS MESSAGES - error messages generated while processing your profile were written to the system log. Correct the errors and resubmit the command.

For the VARY DROP command, the following are user responses for the possible *reasons*:
- INCORRECT CONNECTION NUMBER *number* - issue the onetstat command with the -c option to find the correct connection number. Specify a correct connection number and resubmit the command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.
- INTERNAL COMMAND *number* FAILED ERRNO=*errno* ERRNO2=*errno2* - An internal command was issued to TCP/IP. Report the error to the IBM Software Support Center.

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

© Copyright IBM Corp. 1994, 2002

errno2 is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Programmer Response: None.

Source Data Set: EZACFC00, EZACFMS1, EZACFYAC

Procedure Name: parseCmd

EZZ0060I PROCESSING COMMAND *cmd*

Explanation: TCP/IP is about to process the command.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFC00

Procedure Name: parseCmds

EZZ0150I CONFIGURATION: SIGACTION() FAILED FOR *signal* **:** *reason*

Explanation: The Configuration component encountered an error attempting to setup the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the subagent will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

SIGABND
handler controls error reporting and cleanup functions when an abend occurs. If sigaction fails for SIGABND and an abend occurs, trace information about the abend will be lost and certain resources might not be properly cleaned up. The Configuration component might not be automatically restarted.

SIGTERM
handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, the Configuration component will not be automatically restarted when a SIGTERM is received.

SIGPIPE
handler controls cleanup of resources during termination. If sigaction fails for SIGPIPE, the configuration component will not be automatically restarted when a SIGPIPE is received.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0151I CONFIGURATION: RESTART SCHEDULED

Explanation: An attempt was made to automatically restart the Configuration component following a severe error which caused the Configuration component to terminate. This message is preceded by an error message indicating why the Configuration component was terminated.

System Action: An automatic restart of the Configuration component is attempted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMMN

Procedure Name: cfTermHandler, cfAbendHandler

EZZ0152I CONFIGURATION: SHUTDOWN IN PROGRESS

Explanation: The Configuration component is permanently ending due to an error too severe to attempt an automatic restart.

System Action: The Configuration component is ended.

User or Operator Response: Contact your system programmer.

System Programmer Response: This message is preceded by one or more error messages indicating the error that caused the Configuration component to be ended.

Source Data Set: EZACFMMN

Procedure Name: cfAbendHandler

EZZ0153I CONFIGURATION: POSSIBLE RESTART LOOP DETECTED

Explanation: An error occurred in the Configuration component that caused a restart to be attempted. The restart processing determined that the configuration component had already been restarted multiple times within a short time span and therefore ended the restart processing to prevent a restart loop from occurring.

System Action: The Configuration component is ended. In order to restart the Configuration component, TCP/IP must be stopped and restarted.

User or Operator Response: This message will be preceded by several error messages which will indicate the errors that occurred in the Configuration component that caused the Configuration component to be restarted. No new configurations will be accepted until TCP/IP is stopped and restarted.

System Programmer Response: This message will be preceded by several error messages which will indicate the errors that occurred in the Configuration component to cause the Configuration component to be restarted. Correct the errors indicated by the previous error messages, then stop and restart TCP/IP to restart the Configuration component.

Source Data Set: EZACFMMN

Procedure Name: cfAbendHandler

EZZ0154I CONFIGURATION: UNABLE TO OPEN MESSAGE CATALOG ″cfmsg.cat″ - *errmsg rc/rsn*

Explanation: The configuration component was unable to open the configuration message catalog ″cfmsg.cat″ in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be ″NLSPATH=/usr/lib/nls/msg/%L/%N″.

errmsg describes the Return Code.

rc is the decimal Return Code returned by catopen(). Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

rsn is the hexadecimal Reason Code returned by catopen(). Reason Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

System Action: The configuration component will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If you want to use the external message catalog, contact your system programmer to correct the error. If the default messages are acceptable, no action is necessary.

System Programmer Response: Install the current level of the Configuration message catalog then restart TCPIP to gain access to the message catalog.. If the problem persists, contact the IBM Software Support Center.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0155I CONFIGURATION: UNABLE TO OPEN *ddname* **DD -** *errno*

Explanation: The configuration component attempted to open the data set specified by the indicated DD statement but was unable to open it.

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

System Action: The data set will not be opened.

User or Operator Response: If use of the *ddname* DD statement is required, correct the indicated error. If writing messages to the control log using syslog() is acceptable, no action is necessary.

System Programmer Response: If use of the data set specified on *ddname* is desired, correct the indicated error. If writing messages to the control log using syslog() is acceptable, no action is necessary.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0156I INITIAL PROFILE HANDLING RESULTED IN ERRORS: TERMINATING TCPIP

Explanation: An internal error occurred when handling the initial profile. This error prevents TCP/IP from continuing. TCP/IP generated error messages to the system log. See these messages to determine the reason why TCP/IP is being terminated.

User or Operator Response: Error messages generated while handling the initial profile were written to the system log. Contact your system programmer.

System Programmer Response: Error messages generated while handling the initial profile were written to the system log.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0157I CONFIGURATION: *state*

Explanation: The configuration component might have terminated because of an abend or a SIGTERM condition.

The SIGPIPE handler controls cleanup of resources during termination. If sigaction fails for SIGPIPE, the configuration component will not be automatically restarted when SIGPIPE is received.

Depending on the severity of the error, TCP/IP might continue without the configuration component being active or TCP/IP might terminate.

state describes the state of the configuration component if a termination occurred.

System Action: TCP/IP continues or is terminated.

User or Operator Response: The configuration component terminated. Contact your system programmer.

System Programmer Response: If *state* was **UNABLE TO GET HOSTNAME**, ensure that VMCF is fully initialized before starting TCP/IP. Avoid sharing TCPIP.DATA files between JES nodes. Ensure that there is a separate TCPIP.DATA file for each TCP/IP.

Source Data Set: EZACFMMN

Procedure Name: doErrorRecovery, main

EZZ0158I SELECTEX FAILED: errno=*errno* **errnojr=***errnojr*

Explanation: An internal error occurred in TCP/IP configuration processing.

errno is the hexadecimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The Configuration component will try to take a dump of the TCP/IP address space and then abend.

User or Operator Response: Contact your system programmer.

System Programmer Response: Report the error to the IBM Software Support Center.

Source Data Set: EZACFMMN

Procedure Name: main

EZZ0160I CONFIGURATION: MESSAGE CATALOG FOUND ON PATH *path* **WITH TIMESTAMP** *cat_timestamp* **IS OUT OF DATE - MODULE TIMESTAMP** *mod_timestamp* **- USING DEFAULT MSGS**

Explanation: The configuration component determined that the message catalog ″cfmsg.cat″ found on *path* is out of date. The time stamp found in the message catalog is for an earlier time than that needed by the configuration component load module.

path is the HFS path for the message catalog.

cat_timestamp is the message catalog time stamp in the format: yyyy ddd hh:mm UTC.

mod_timestamp is the message catalog time stamp that was used when the configuration component modules were compiled in the format: yyyy ddd hh:mm UTC.

System Action: The configuration component will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If you want to use the external message catalog, contact your system programmer to correct the error.

If the default messages are acceptable, no action is necessary.

System Programmer Response: An out-of-date message catalog can be caused by any of the following conditions:

- The wrong HFS file system was mounted.
- The NLSPATH environment variable was pointing to an old catalog.
- The service update for the new catalog failed.

If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: main

Procedure Name: EZACFMMN

EZZ0161I AUTOLOG: MESSAGE CATALOG FOUND ON PATH *path* **WITH TIMESTAMP** *cat_timestamp* **IS OUT OF DATE - MODULE TIMESTAMP** *mod_timestamp* **- USING DEFAULT MSGS**

Explanation: The autolog task determined that the message catalog ″cfmsg.cat″ found on *path* is out of date. The time stamp found in the message catalog is for an earlier time than that needed by the autolog load module.

path is the HFS path for the message catalog.

cat_timestamp is the message catalog time stamp in the format: yyyy ddd hh:mm UTC.

mod_timestamp is the message catalog time stamp that was used when the autolog module was compiled in the format: yyyy ddd hh:mm UTC.

System Action: The autolog task will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If you want to use the external message catalog, contact your system programmer to correct the error.

If the default messages are acceptable, no action is necessary.

System Programmer Response: An out-of-date message catalog can be caused by any of the following conditions:

- The wrong HFS file system was mounted.
- The NLSPATH environment variable was pointing to an old catalog.
- The service update for the new catalog failed.

If the problem cannot be corrected, then contact the IBM Software Support Center.

Source Data Set: main

Procedure Name: EZACFALG

EZZ0300I Opened *ftype* **file** *fname*

Explanation: The file was opened for the specified type of processing.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPPR

Procedure Name: findInitProfile, openObeyFile, processInclude

---

EZZ0301I Dataset name *dsn* **incorrect**

Explanation: The data set name does not conform to MVS data set naming conventions.

System Action: Configuration processing ends for that data set.

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

System Programmer Response: None.

Source Data Set: EZACFD00, EZACFPPR

Procedure Name: openObeyFile, processInclude

EZZ0302I End of File encountered

Explanation: The end of file was found before all configuration processing could be completed.

System Action: Profile processing ends.

User or Operator Response: Correct the previous statement and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0303I *ftype* **file contains errors**

Explanation: The indicated file contains errors.

System Action: Configuration processing continues.

User or Operator Response: Correct the errors in the data set and continue.

System Programmer Response: None.

Source Data Set: EZACFYAC, EZACFPPR

Procedure Name: processInclude, handleInitialProfile, processObeyFile

---

EZZ0304I Resuming processing of file *dsn*

Explanation: Completed processing of an include file. The original configuration file continues.

System Action: The file that contained the include statement continues.

User or Operator Response: Correct the problems in the include file and rerun.

System Programmer Response: None.

Source Data Set: EZACFYAC

Procedure Name: processInclude

EZZ0305I Cannot open file *dsn*

Explanation: The indicated file cannot be opened.

System Action: Profile processing cannot be done for the data set.

User or Operator Response: Correct the open failure and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPPR

Procedure Name: processInclude, processObeyFile

EZZ0306I Attempted recursive include of *dsn* **on line** *lineno*

Explanation: A file cannot be included multiple times for the same configuration file processing.

System Action: The file is not included.

User or Operator Response: Correct the include statement and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPPR

Procedure Name: processInclude

EZZ0307I CONFIGURATION: An internal error occurred, the reason code is *reason_code*

Explanation: An error occurred in the configuration component.

reason_code is one of the following:

-1 Selectex call failed for the Autolog task. The Autolog task will be shut down.

-4 Internal error occurred. This will be one of the following:
- The call to get the telnet socket/port failed.
- Telnet error when trying to retrieve the stack jobname from TSEB (no TSEB found.)

-5 Should not occur - internal Telnet error.

-15 Storage obtain failed.

-17 Header contained invalid data. This might be a verson or request type that is not valid.

-18 Header length invalid.

-20 Bad contact type in header. The contact value must be either INIT_CONTACT, RECONTACT, or RESYNC. The connection is closed.

-22 Buffer too small to receive data. Input buffer is not large enough to be a protocol header.

-23 Buffer too small to be valid. Output buffer is not large enough for a response header.

-24 Bad subtype in buffer. Subtype in response does not match subtype in request.

-25 Address less than or equal to zero or Telnet client state not valid to send QUIESCE.

-26 Data in response is invalid. Telnet error in QUIESCE response.

-27 Socket descriptor too large.

-28 Socket descriptor too large.

-50 User tokens were not equal on a RESYNC request.

-52 Maximum number of retries reached for profile processing on RECONTACT.

-60 Cannot fine TSEB in main entry point for configuration subtask. Configuration task will be shut down.

-61 Cannot fine TSDB in main entry point for configuration subtask. Configuration task will be shut down.

-62 Cannot fine TSDX in main entry point for configuration subtask. Configuration task will be shut down.

-63 Socket call failed. Error trying to open the socket to the stack when processing a start or stop device.

-64 Storage obtain failed. Error when attempting to get a buffer for ioctl when processing a start or stop device.

-65 Storage obtain failed. Error when attempting to get a buffer for ioctl when processing initial device.

-66 Storage obtain failed. Error when attempting to get a buffer for ioctl when processing obeyfile.

-67 Setibmopt failed while processing a start or stop device. In order to have more than one TCP/IP stack connected to z/OS UNIX, the z/OS UNIX must be configured for Common INET (CINET.) Processing is terminated

-68 Autolog initialization failed.

-69 Cannot find TSEB in main entry point for Autolog subtask. Autolog task will be shut down.

-70 Cannot find TSDB in main entry point for Autolog subtask. Autolog task will be shut down.

-71 Cannot find TSDX in main entry point for Autolog subtask. Autolog task will be shut down.

-72 Autolog error recovery failed.

System Action: TCP/IP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the error described by error_code. Report Internal errors to the IBM Software Support Center.

Source Data Set: EZACFMMN, EZACFPAR

Procedure Name: main, handleInitialProfile, processObeyFile

EZZ0309I Profile processing beginning for *dsn*

Explanation: Profile processing beginning.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFYAC

Procedure Name: parseFile

EZZ0310I File *dsn* **contains no statements**

Explanation: The file contains no statements.

System Action: None.

User or Operator Response: Ensure that the data set name was correctly entered and contains valid statements.

System Programmer Response: None.

Source Data Set: EZACFYAC

Procedure Name: parseFile

EZZ0311I The *statement* **statement on line** *lineno* **is obsolete**

Explanation: The statement is obsolete. This usually indicates that an unmodified profile from a prior release is being used.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFYAC

Procedure Name: parseFile

EZZ0312I The *statement or argument* **on line** *lineno* **contains an incorrect value** *value*

Explanation: An incorrect value was specified.

System Action: Profile processing continues. However, the statement or argument is not defined properly and will be ignored.

For *read_buffers* or *write_buffers* of a CDLC device, the amount of buffer space used for either the read or write size must be less than 64K. Therefore for a buffer size of 4096, a maximum of 15 buffers can be reserved. For a size of 2048, a maximum 31 buffers can be used. For a size of 1024, a maximum 63 buffers can be used.

This message can be issued when defining a supernet route on the GATEWAY or BEGINROUTES statement if IPCONFIG VARSUBNETTING has not yet been set.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFMS1, EZACFATM

Procedure Name: parseFile

EZZ0313I The *option* **option in** *statement* **statement on line** *lineno* **is not supported in this release**

Explanation: The option is not supported in this release.

System Action: Profile processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFPPT, EZACFATM

Procedure Name: parseFile

EZZ0314I Extraneous parameter *parm* **found on line** *lineno*

Explanation: The parameter was used incorrectly.

System Action: The parameter is ignored.

User or Operator Response: Correct the parameter and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0315I D...TELNET,HNGROUP <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,ID=XHNGNM,<SUMM|DET>> <,MAX=XNN|*>

Explanation: This message is the result of the HELP DISPLAY HNGROUP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0316I Profile processing complete for file *dsn*

Explanation: Profile processing complete for the specified data set.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFYAC

Procedure Name: parseFile

EZZ0318I *value* **was found on line** *lineno* **and** *parameter* **was expected**

Explanation: The expected parameter was not found.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFATM, EZACFPAL, EZACFPPT, EZACFPIT

Procedure Name: parseFile

EZZ0319I The *Entry* **entry value** *value* **is an incorrect** *parameter* **parameter on** *statement* **statement on line** *lineno*

Explanation: The value is incorrect.

System Action: The value is ignored. Profile processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0320I *Command* **value** *val* **on line** *lineno* **is too long**

Explanation: The value is too long.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACDSTR, EZACFPPR, EZACFPPT

Procedure Name: parseFile

EZZ0321I Internal command *number* **failed errno=***errno* **errnojr=***errnojr* **on line** *lineno*

Explanation: An internal error occurred in TCP/IP configuration processing.

errno is the hexadecimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The statement fails.

User or Operator Response: Contact your system programmer.

System Programmer Response: Report the error to the IBM Software Support Center.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0322I *keyword* **value** *value* **not valid on line** *lineno***,** *replacement* **used**

Explanation: The value is incorrect, the replacement value (which might be either the default value or the closest allowed value) is used instead.

System Action: The replacement value is used. Processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFPIT, EZACFATM, EZACFPAL, EZACFPPT

Procedure Name: parseFile

EZZ0323I *Statement* **statement on line** *lineno* **had no entries**

Explanation: The statement had no entries. This might have been caused by a HOME, GATEWAY, BEGINROUTES, AUTOLOG, NETACCESS, BSDROUTINGPARMS, or TRANSLATE statement that had a syntax error in the first entry.

If this is a HOME statement and it is the first HOME statement in a profile, the NULL HOME statement will delete all existing HOME entries. Similarly, if this is the first GATEWAY, BEGINROUTES, AUTOLOG, NETACCESS, or TRANSLATE statement, all existing entries of that type are deleted. If this is not the desired effect, update the profile and rerun it.

System Action: None.

User or Operator Response: Update the file and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR, EZACFPAL

Procedure Name: parseFile

EZZ0324I Unrecognized statement *statement* **found on line** *lineno*

Explanation: The statement is unrecognized.

This error might have been caused by a problem on the previous line or command such as:
- Incomplete parameter information.
- Extraneous parameter information.
- Mistyped keyword parameters.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFYAC

Procedure Name: parseFile

EZZ0325I Initial profile could not be found

Explanation: The initial profile could not be found.

System Action: Profile processing ends. TCP/IP ends.

User or Operator Response: Update the files used for initial profile processing. Consult the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: TCP/IP ends.

Source Data Set: EZACFPPR

Procedure Name: HandleInitialProfile

EZZ0326I *statement1* **conflicts with** *statement2* **value** *value* **used on line** *lineno*

Explanation: The value is incorrect.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFATM, EZACFMS1, EZACFPPT

Procedure Name: parseFile

EZZ0327I *Type name* **on line** *lineno* **is already defined**

Explanation: The name is already defined.

For ATMLIS type statements, it is possible to change the *ATMLIS Options*. But the *subnet_value subnet_mask* cannot be changed. To change these values the ATMLIS must be deleted then redefined.

When type is LINK NAME, this message is displayed when a link name is being reassigned to a new device after being deleted. A link name cannot be reassigned to a new device while TCPIP is active.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0328I *Type name* **on line** *lineno* **has not been defined or has been deleted**

Explanation: The name was not defined or was deleted.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0329I Linkname *Linkname* is not defined but is used for BSDRouting on line *lineno*

Explanation: Linkname *Linkname* is not defined but is used for BSDRouting on line *lineno*.

System Action: The statement is not processed. All BSDRouting statements that follow are not processed.

User or Operator Response: Define the Linkname or remove it from BSDRouting Parms.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0330I On line *lineno* an attempt was made to define more *type1* to *type2 devname* than is allowed by the device

Explanation: The device specified has more links already defined than the device type supports. For example, CTC devices can have at most one link. LCS devices can have a link statement for each adapter.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0331I No home address assigned to link *linkname*

Explanation: The link was defined but no home address was given for it. No TCP/IP traffic can flow over this link.

System Action: TCP/IP continues.

User or Operator Response: Assign a home address for the link using the HOME statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0332I DD:PROFILE not found. Continuing profile search

Explanation: Either there was no //PROFILE DD JCL card specified in the TCP/IP cataloged procedure or the //PROFILE DD JCL card specified could not be opened.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPPR

Procedure Name: parseFile

EZZ0333I Linkname *Linkname* is not defined with BSDROUTINGPARMS information

Explanation: Linkname *Linkname* is not defined with BSDROUTINGPARMS information. If BSDROUTINGPARMS are used, each linkname should be defined once in the BSDROUTINGPARMS statement. The link will get default BSDROUTINGPARMS information.

System Action: TCP/IP continues.

User or Operator Response: Use onetstat -d to determine if the default BSDROUTINGPARMS information is what is desired.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0334I IP forwarding is disabled

Explanation: The status of IP forwarding is displayed. IP forwarding can be disabled by specifying NODATAGRAMFWD on the IPCONFIG statement, or NOFWD on the ASSORTEDPARMS statement.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0335I ICMP *< will | will not >* **ignore redirects**

Explanation: The state of ICMP redirects is displayed.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0336I *<A limit | No limit>* **on incoming UDP datagram queue set**

Explanation: The limit on incoming UDP datagram queue is displayed.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0337I CLAWUSEDOUBLENOP is *< set | cleared >*

Explanation: The status of CLAWUSEDOUBLENOP is displayed.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0338I *< TCP | UDP>* **ports 1 thru 1023** *< are | are not>* **reserved**

Explanation: The status of the TCP or UDP restricted ports is displayed.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0339I More messages were found than can be reported

Explanation: The semantic processing at the end of the profile completes many configuration checks. These include LINKs without HOME statements, BSD routing checks, etc. This message is issued when more of these conditions exist than can be reported.

System Action: TCP/IP continues.

User or Operator Response: Check previous error messages, correct the problems that exist, and rerun the profile to receive the messages that could not be reported during this profile processing.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0340I BSDROUTINGPARMS destination address *dest_addr* **for link** *linkname* **is incorrect**

Explanation: The BSDROUTINGPARM destination address for the specified link is incorrect. If the link is a CTC or any other point to point link, the destination address must be a host address. if the link is not a point to point link, then the destination address must be a network or subnetwork address.

System Action: TCP/IP continues.TCPIP ignores the statement.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0341I Configuration component cannot modify TCP/IP configuration errno=*errno* **errnojr=***errnojr*

Explanation: A prior internal error eliminated the ability of the configuration component to modify the TCP/IP stack.

errno is the hexadecimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: TCP/IP continues with the configuration unchanged. If this is the initial profile, TCP/IP ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The stack configuration remains unchanged. The only way to circumvent the prior internal error problem is to stop TCP/IP and restart it. If the error occurs on the initial profile, the TCP/IP will not start up.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0342I DELETE PORT/PORTRANGE failed on line *lineno*

Explanation: One of the following occurred:
- The port was never defined
- The port was defined, but then deleted
- The port define attributes do not agree with the delete attributes

  For example, a PORTRANGE was used to define the port, but a DELETE PORT was used to delete the port.

System Action: TCP/IP continues.

User or Operator Response: Correct the statement and rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0343I Mismatched *statement* **on line** *lineno*

Explanation: The characteristic used on the specified line conflicts with a previous definition. One of the following occurred:
- A Translate statement specified a network address not supported by the linkname used.
- An ATMARPSV statement specified a pvc name that is defined to a link name that is not defined to the ATMARPSV logical IP subnetwork.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0344I Permanent loopback address 127.0.0.1 specified on line *lineno* **cannot be added to the HOME list**

Explanation: The HOME list entry on the specified line used the permanent loopback IP address 127.0.0.1. The permanent loopback IP address 127.0.0.1 cannot be added to the HOME list.

System Action: TCPIP ignores the statement. TCP/IP continues.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0345I STOPONCLAWERROR is *< enabled | disabled>*

Explanation: The status of STOPONCLAWERROR is displayed. After STOPONCLAWERROR is enabled it cannot be disabled.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0348I Gate list entry number *routenum* **on line** *lineno* **for destination** *destaddr* **is unreachable through interface** *intfaceaddr* **on** *interface*

Explanation: The specified gate list entry within the GATEWAY statement on the line specified is in error. The destination address is unreachable through the specified interface using any known routes.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0349I Firewall support is *status*

Explanation: The status of Firewall support is displayed, where *status* is either: ENABLED or DISABLED.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0350I Sysplex routing support is *status*

Explanation: The status of Sysplex routing support is displayed, where *status* is either: ENABLED or DISABLED.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0351I SourceVipa support is *status*

Explanation: The status of SourceVipa support is displayed, where *status* is either: ENABLED or DISABLED.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0352I Variable Subnetting support is *status* **for OROUTED**

Explanation: The status of Variable subnetting support for OROUTED is displayed, where *status* is either: ENABLED or DISABLED.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0353I Device *devicename* **on line** *lineno* **is** *devstate*

Explanation: An attempt to change the state of a device failed because of the reason stated.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0354I *Type name* **on line** *lineno* **is redefined**

Explanation: The name had been previously defined and the new definition was used in its place.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0355I **D...<NETSTAT>,CONN<,CLIENT=|IPADDR=|PORT=|NOTN3270><,FORMAT=LONG|SHORT>**
D...TELNET,CONN <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,CONN=XCONID>
<,IPPORT=XIPADDR..XPORT> <,LUNAME=XLUNM<,SUMM|DET>> <,LUNAME=XLUNM*>>
<APPL=<XAPPLNM|XAPPLNM*>> <,IPADDR=<XIPADDR|XMASK:XSUBNET>>
<,LUGROUP=XLUGPNM> <,IPGROUP=XIPGPNM> <,PROTOCOL=XPROTMODE> <,MAX=XNN|*>

Explanation: This message is the result of issuing the HELP CONN command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0356I *A* **and** *B* **are mutually exclusive:** *A* **on line** *lineno {accepted|ignored}*

Explanation: Two parameters are mutually exclusive. Accept or ignore statement.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0357I **FOR DISPLAY HELP ENTER D... HELP,HELP**

Explanation: This message is the result of issuing the HELP HELP command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0358I **V...(,OBEYFILE|PKTTRACE|DATTRACE|START| STOP|NETSTAT|TELNET|PURGECACHE)**

Explanation: This message is the result of issuing the HELP VARY command and shows what help is available for the VARY command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0359I V...(OBEYFILE|CMD=O), (DSN=XDSNAME|XDSNAME)

Explanation: This message is the result of the HELP OBEYFILE command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0360I V...PKTTRACE,..(SEE PUBLICATION FOR KEYWORDS)

Explanation: This message is the result of the HELP PKTTRACE command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0361I V...(START|CMD=START),XDEVNAME

Explanation: This message is the result of the HELP START command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0362I V...(STOP|CMD=STOP),XDEVNAME <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE>
V...TELNET,STOP <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE>

Explanation: This message is the result of the HELP STOP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0363I V...DATTRACE,..(SEE PUBLICATION FOR KEYWORDS)

Explanation: This message is the result of the HELP DATTRACE command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0364I V...(DROP|CMD=DROP)(,XCONNID|CONN=XCONNID)

Explanation: This message is the result of the DISPLAY HELP DROP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0365I D...TELNET,LINKNAME <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,MAX=XNN|*>

Explanation: This message is the result of the HELP LINKNAME command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0366I V...TELNET,ACT,XLUNAME

Explanation: This message is the result of the HELP ACT command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0367I V...TELNET,INACT,XLUNAME

Explanation: This message is the result of the HELP INACT command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0368I V...TELNET,QUIESCE <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE>

Explanation: This message is the result of the HELP QUIESCE command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0369I V...TELNET,RESUME <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE>

Explanation: This message is the result of the HELP RESUME command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0371I D...(NETSTAT|TELNET|HELP|DISPLAY|VARY|OMPROUTE|SYSPLEX|STOR)

Explanation: This message is the result of the HELP DISPLAY command and shows what help is available for the display command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0372I D...NETSTAT(,ACCESS,ALLCONN|ARP|BYTEINFO|CACHINFO|CONFIG|CONN|DEVLINKS|DROP|HOME|IDS|ND|PORTLIST|ROUTE|SOCKETS|STATS|VCRT|VDPT|VIPADCFG|VIPADYN)

Explanation: This message is the result of the HELP NETSTAT command and shows what help is available for Netstat commands.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0373I D...TELNET(,APPL|CONNECTION|DEFAULTS|DEVICETYPE|HNGROUP| INACTLUS|IPGROUP|LINKNAME|LUGROUP|LUMAP|PARMSGROUP|PARMSMAP| PROFILE|WHEREUSED|WLM) V...TELNET(,ACT|INACT|QUIESCE|RESUME|STOP)

Explanation: This message is the result of the HELP TELNET command and shows what help is available for Telnet commands.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0374I D...NETSTAT,ALLCONN<,CLIENT=|IPADDR=|PORT=|NOTN3270><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP ALLCONN command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0375I D...NETSTAT,ARP<,XNETADDR>

Explanation: This message is the result of the DISPLAY HELP ARP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0376I D...NETSTAT,BYTEINFO<,IDLETIME><,CLIENT=|NOTN3270><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP BYTEINFO command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0377I D...NETSTAT,CONFIG<,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP CONFIG command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0378I D...NETSTAT,DEVLINKS<,INTFNAME=><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP DEVLINKS command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0379I D...NETSTAT,ROUTE<,RSTAT|IQDIO><,ADDRTYPE=IPV4|IPV6><,DETAIL><IPADDR=><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP ROUTE command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0380I D...NETSTAT,PORTLIST<,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP PORTLIST command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0381I D...NETSTAT,SOCKETS<,CLIENT=|IPADDR=|PORT=|NOTN3270><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP SOCKETS command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0382I D...NETSTAT,HOME<,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP HOME command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0383I D...TELNET,APPL <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,TYPE=<ACT|PROF>| <ID=XAPPL,<SUMM|DET>>> <,MAX=XNN|*>

Explanation: This message is the result of the HELP APPL command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0384I D...TELNET,DEFAULTS <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,MAX=XNN|*>

Explanation: This message is the result of the HELP DEFAULTS command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0385I D...TELNET,DEVICETYPE <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,MAX=XNN|*>

Explanation: This message is the result of the HELP DEVICETYPE command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0386I D...TELNET,IPGROUP <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,ID=XIPGNM,<SUMM|DET>> <,MAX=XNN|*>

Explanation: This message is the result of the HELP IPGROUP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0387I D...TELNET,LUGROUP <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,ID=XLUNAM,<SUMM|DET>> <,MAX=XNN|*>

Explanation: This message is the result of the HELP LUGROUP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0388I D...TELNET,LUMAP <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,ID=XLUNAM|XPRTNM <,SUM|DET>> <,MAX=XNN|*>

Explanation: This message is the result of the HELP LUMAP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0389I D...TELNET,PROFILE <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,SUMMARY|DETAIL> <,MAX=XNN|*>

Explanation: This message is the result of the HELP PROFILE command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0390I D...TELNET,WLM<,MAX=XNN|*>

Explanation: This message is the result of the HELP WLM command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0391I D...TELNET,WHEREUSED <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE>

Explanation: This message is the result of the HELP WHEREUSED command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0392I CANNOT START FIREWALL AFTER TCPIP IS ACTIVE.

Explanation: Firewall can only be started from an initial profile.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Halt and restart TCP/IP to start Firewall function.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0393I FIREWALL MODULE ICAFWALL LOAD FAILED

Explanation: The Firewall program load module ICAFWALL could not be loaded.

System Action: TCP/IP Initialization will not complete successfully.

User or Operator Response: None.

System Programmer Response: Correct the problem and restart TCP/IP.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0394I FIREWALL SETUP FAILED RETURN *rc* **REASON** *rsn*

Explanation: Firewall initialization failed.

System Action: TCP/IP Initialization will not complete successfully.

User or Operator Response: None.

System Programmer Response: Refer to Firewall documentation for guidance.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0395I *statement name* **ON LINE** *lineno* **FAILED BECAUSE** *reason*

Explanation: The command failed for the reason indicated.

statement is the profile statement that failed.

name is the name specified on the profile statement.

lineno is the line number in the profile data set where the statement was found.

reason is one of the following:

A LINK REFERENCES THE ATMLIS
An ATM LINK statement referenced this ATMLIS.

AN ATMARPSV REFERENCES THE ATMLIS
An ATMARPSV statement referenced this ATMLIS.

A VIPA DEVICE CANNOT BE STOPPED OR STARTED
A VIPA device is started when defined and can never be stopped.

DEVICE HAS A LINK DEFINED
The device has a link defined.

DEVICE IS ACTIVE
The device is active.

INTERFACE IS ACTIVE
The interface is not in a valid state to be deleted. The interface must be inactivated.

INTERFACE IS DEFINED AS A SOURCE VIPA INTERFACE
The interface is specified on the SOURCEVIPAINTERFACE keyword for another interface and therefore cannot be deleted.

LINK IS ACTIVE
The link is active.

LINK STATE NOT VALID
The device or link is not in a valid state to be deleted.

LINK TYPE NOT VALID
The link type for the specified name is not valid for the statement.

THE ATMARPSV IS BUSY
The ATMARPSV cannot be deleted when it is in use.

THE ATMPVC IS USED BY AN ATMARPSV
The ATMPVC cannot be deleted until the ATMARPSV that specifies it is deleted.

THE LINK HAS PVCS DEFINED
An ATMPVC statement has a PVC to this link defined.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: See the following based on reason indicated:

A LINK REFERENCES THE ATMLIS
The ATM LINK must be deleted before the ATMLIS can be deleted. The VARY TCPIP,,OBEYFILE command can be issued to delete the ATM LINK.

AN ATMARPSV REFERENCES THE ATMLIS
ATMARPSV must be deleted before the ATMLIS can be deleted. The VARY TCPIP,,OBEYFILE command can be issued to delete the ATMARPSV.

A VIPA DEVICE CANNOT BE STOPPED OR STARTED
No action.

DEVICE HAS A LINK DEFINED
The link must be deleted before the device can be deleted. The VARY TCPIP,,OBEYFILE can be used to delete the link.

DEVICE IS ACTIVE
The device must be stopped before it can be deleted. The VARY TCPIP,,STOP command can be issued to stop the device.

LINK STATE IS NOT VALID
The link is in use. In the case of a link delete attempt, the home IP address in the HOME statement must be removed. The VARY TCPIP,,OBEYFILE can be issued to update the HOME list excluding the home IP address associated with the link to be deleted.

THE ATARPSV IS BUSY
No action.

THE ATMPVC IS USED BY AN ATMARPSV
ATMARPSV must be deleted before ATMPVC can be deleted. The VARY TCPIP,,OBEYFILE command can be issued to delete the ATMARPSV.

THE LINK HAS PVCS DEFINED
The PVC must be deleted before the link can be deleted. The VARY TCPIP,,OBEYFILE command can be issued to delete the PVC.

Source Data Set: EZACFMS1

Procedure Name: parseCmd

EZZ0396I SNMP SUBAGENT *< ALREADY STARTED | ALREADY STOPPED >*

Explanation: The state of the SNMP subagent was not changed. Either SACONFIG ENABLED was specified and the subagent was already active, or SACONFIG DISABLED was specified but the subagent was not active.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0397I ATMLIS *lisname* **OPTIONS ON LINE** *lineno* **ARE UPDATED.**

Explanation: The ATMLIS options entered on the line specified were updated.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0398I *link* **SUBNET MASK** *value* **ON LINE** *lineno* **IS NOT IN CIDR FORMAT**

Explanation: The specified link's mask address value is not in Classless Inter-Domain Routing (CIDR) format. CIDR says that a mask can be consecutive bits of ones then zero or more zeros. It cannot then have more ones following.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0400I **TELNET/VTAM (SECOND PASS) BEGINNING FOR FILE** *filename*

Explanation: The module will begin to process the specified file.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPTV

Procedure Name: Telnet/Vtam parser

EZZ0401I *Condition* **IN FILE:** *filename* **ON LINE:** *lineno* **AT:** *token*

Explanation: The specified event occurred while processing the file. The token shown was found at the specified line number in the named file. The actual event might be associated with this or the previous token.

System Action: Profile processing continues.

User or Operator Response: If an error is indicated, correct the statement.

System Programmer Response: None.

Source Data Set: EZACFPTV, EZACFLEX, EZACFPPR

Procedure Name: Telnet/Vtam parser

EZZ0402I *Num* **correct statement(s) were attempted from file:** *filename*

Explanation: The module found the indicated number of syntactically correct statements in the indicated file.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPTV

Procedure Name: Telnet/Vtam parser

EZZ0403I **TELNET/VTAM (SECOND PASS) COMPLETE FOR FILE:** *filename*

Explanation: The module finished processing the specified file.

System Action: Profile processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPTV

Procedure Name: Telnet/Vtam parser

EZZ0600I *port_type port_num* **CONFLICTS WITH PORT RESERVATION**

Explanation: The port is already reserved.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0601I **ipInReceives** *recv_count* **ipOutRequests** *out_count* **ipForwDatagrams** *for_count*

Explanation: This is a result of GLOBALCONFIG or ASSORTEDPARMS TCPIPSTATISTICS.

recv_count is the number of IP requests received.

*out_coun*t is the number of IP requests sent.

for_count is the number of IP datagrams forwarded.

System Action: TCP/IP is being shutdown.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFGIO

Procedure Name: parseFile

EZZ0602I **ipReasmReqds** *req* **ipReasmFails** *fails* **ipFragCreates** *creates* **ipFragFails** *frag_fails*

Explanation: The number of IP reassembly requests received and failed, and IP fragments created and failed are displayed. This is a result of GLOBALCONFIG or ASSORTEDPARMS TCPIPSTATISTICS parameter.

System Action: TCP/IP is being shutdown.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFGIO

Procedure Name: parseFile

EZZ0603I **icmpInMsgs** *icmp_in* **icmpOutMsgs** *icmp_out* **udpInDatagrams** *udp_in* **udpOutDatagrams** *udp_out*

Explanation: The number of ICMP redirect messages received and sent, and UDP datagrams received and sent are displayed. This is a result of GLOBALCONFIG or ASSORTEDPARMS TCPIPSTATISTICS parameter.

System Action: TCP/IP is being shutdown.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFGIO

Procedure Name: parseFile

EZZ0604I **tcpInSegs** *tcp_in* **tcpOutSegs** *tcp_out* **tcpRetransSegs** *tcp_re*

Explanation: The number of tcp segments received, segments sent and segments retransmitted are displayed. This is a result of GLOBALCONFIG or ASSORTEDPARMS TCPIPSTATISTICS parameter.

System Action: TCP/IP is being shutdown.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFGIO

Procedure Name: parseFile

EZZ0605I LINK *linkname* **HAS INCOMPATIBLE BSDROUTINGPARMS SUBNET MASK** *mask*, **DESTINATION ADDRESS** *dest* **AND HOME IPADDRESS** *home*

Explanation: The link specified has incompatible subnet mask, destination address and HOME ipaddress.

System Action: TCP/IP continues.

User or Operator Response: Correct the BsdRoutingParms and HOME statements. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0606I VARSUBNETTING CANNOT BE TURNED OFF. LINK *linkname* **HAS BSDROUTINGPARMS SUPER SUBNET MASK** *mask* **AND HOME IPADDRESS** *home*

Explanation: The link specified has a super subnet mask. VARSUBNETTING cannot be turned off with any links having a ClassA Home address and less than 255.0.0.0 mask, or a ClassB Home address and less than 255.255.0.0 mask, or a ClassC Home address and less than 255.255.255.0 mask.

- A ClassA Home address is 0.0.0.0 thru 127.255.255.255
- A ClassB Home address is 128.0.0.0 thru 191.255.255.255
- A ClassC Home address is 192.0.0.0 thru 233.255.255.255

System Action: TCP/IP continues.

User or Operator Response: Correct the BsdRoutingParms and HOME statements. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0607I VARSUBNETTING CANNOT BE TURNED OFF. ROUTE *type network link pktsize gwaddr* **IS A SUPER NETWORK ROUTE**

Explanation: The specified route is a super subnet route. VARSUBNETTING cannot be turned off with any routes having a ClassA destination address and less than 255.0.0.0 mask, or a ClassB destination address and less than 255.255.0.0 mask, or a ClassC destination address and less than 255.255.255.0 mask.

- A ClassA address is 0.0.0.0 thru 127.255.255.255
- A ClassB address is 128.0.0.0 thru 191.255.255.255
- A ClassC address is 192.0.0.0 thru 233.255.255.255

Note that the type value can be:
- GATEWAY
- ICMP
- OMPROUTE
- ROUTED or OMPROUTE

System Action: TCP/IP continues.

User or Operator Response: Correct the route. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0608I **LINK** *lnkname* **HAS LIS** *lisname* **WITH ATMLIS SUBNET MASK/VALUE** *submsk/subval* **THAT DOES NOT AGREE WITH HOME ADDRESS** *ipaddr*

Explanation: The specified link's LIS was defined with the ATMLIS subnet and value. But the specified HOME address used for this LINK is not within that network. The LINK will not be available for SVC traffic.

System Action: TCP/IP continues.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0609I *stmttype stmtname* **IS UNREFERENCED BY ANY** *stmttype2* **STATEMENT. IT WAS DELETED**

Explanation: The specified ATMLIS or ATMARPSV statement is unreferenced. The ATMLIS was defined but never used on a LINK statement. Therefore, that ATMLIS and all ATMARPSV's that reference that ATMLIS cannot be used by TCP/IP. Therefore the ATMLIS or ATMARPSV was deleted from TCP/IP.

System Action: TCP/IP continues.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0610I *keyword* **VALUE** *value* **NOT VALID ON LINE** *lineno*, *replacement* **USED**

Explanation: The value is incorrect, the replacement value (which might be either the default value or the closest allowed value) is used instead.

System Action: The replacement value is used. Processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0611I *type1 val1* **on line** *lineno* **HAS BEEN USED PREVIOUSLY FOR** *type2 val2*

Explanation: The value used on the specified line cannot be used again for the specified type. For example, two LINKs on an ATM DEVICE cannot use the same LIS.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0612I **HOME ADDRESS** *val1* **FOR LINK** *lnkname* **ON LINE** *lineno* **REPLACES THE PREVIOUS ADDRESS**

Explanation: The home statement on the specified link redefines a previously defined IP address for the specified link name. The home address replaces the old value.

System Action: Profile processing continues.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0613I TCPIPSTATISTICS IS *status*

Explanation: The status of GLOBALCONFIG/ASSORTEDPARMS TCPIPSTATISTICS is displayed, where *status* is either: ENABLED or DISABLED.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0614I TCPCONFIG TCPMAXRCVBUFRSIZE VALUE *value* **IS LESS THAN TCPRCVBUFRSIZE VALUE** *specified* **-** *defaults* **USED.**

Explanation: TCPCONFIG TCPMAXRCVBUFRSIZE value specified is less than the TCPRCVBUFRSIZE value that is specified.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0615I MULTIPATH SUPPORT IS DISABLED

Explanation: The status of Multipath support is displayed.

If Multipath support is disabled and there are multiple equal-cost paths to a destination, then TCP/IP will use the first path found for all IP packets to that destination.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0616I TCPCONFIG *(TCPMAXRCVBUFRSIZE | TCPRCVBUFRSIZE)* **VALUE SPECIFIED IS** *(LESS | GREATER)* **THAN** *(TCPRCVBUFRSIZE | TCPMAXRCVBUFRSIZE)* **VALUE DEFINED - THE VALUE IS IGNORED.**

Explanation: Either TCPCONFIG TCPMAXRCVBUFRSIZE value is less than TCPRCVBUFRSIZE value that is defined, or TCPCONFIG TCPRCVBUFRSIZE value is greater than TCPMAXRCVBUFRSIZE value that is defined.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0617I HOME ADDRESS *val1* **FOR LINK** *lnkname* **ON LINE** *lineno* **IS A RESERVED IP ADDRESS**

Explanation: The home statement on the specified line uses a reserved IP address for the specified link name. The home address is not assigned.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Migration*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0619I LINK *lnkname* **USES DUPLICATE HOME ADDRESS** *addr*

Explanation: Two or more interfaces have the same IP address. If both are active at the same time, the results are unpredictable.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Migration*. Change one of the interfaces to a different HOME address.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0620I LINK *lnkname* **ON DEVICE** *devname* **USES DUPLICATE ADAPTER NUMBER** *number*

Explanation: Two or more links have the same adapter number.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Migration*. Change one of the adapter numbers to a different number before starting the device to avoid a start device error.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0621I AUTOLOG FORCING *jobname*, **REASON:** *reason*

Explanation: Autolog processing determined that jobname is currently active and will issue MVS CANCEL command for the jobname.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Migration*.

System Programmer Response: None.

Source Data Set: EZACFALG

Procedure Name: parseFile

EZZ0622I AUTOLOG FORCING *jobname* **- DUMP PROVIDED, REASON:** *reason*

Explanation: Autolog processing determined that jobname is currently active and will issue MVS CANCEL command for the jobname. CANCELWITHDUMP was coded on the AUTOLOG statement, and an MVS dump is provided with the CANCEL.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Migration*.

System Programmer Response: None.

Source Data Set: EZACFALG

Procedure Name: parseFile

EZZ0623I PATH MTU DISCOVERY SUPPORT IS *status*

Explanation: The status of PATH MTU Discovery support is displayed, where *status* is either: ENABLED or DISABLED. If PATHMTUDISCOVERY was coded on the IPCONFIG statement, then support status is enabled. If Path MTU Discovery support is enabled, then TCP/IP will dynamically discover the Path MTU (PMTU), which is the minimum of the MTUs of each hop in the path, necessary to prevent fragmentation of datagrams sent on that path. If NOPATHMTUDISCOVERY was coded on the IPCONFIG statement, then support status is disabled and large datagrams might be fragmented.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0624I DYNAMIC XCF DEFINITIONS ARE *status*

Explanation: The status of Dynamic XCF definitions are displayed, where *status* is either: ENABLED or DISABLED. If DYNAMICXCF was coded on the IPCONFIG statement, then support status is enabled. If NODYNAMICXCF was coded on the IPCONFIG statement, then support status is disabled.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0625I IPCONFIG DYNAMICXCF FAILED BECAUSE XCF DYNAMIC DEVICES STILL ACTIVE

Explanation: The IP address specified on the DYNAMICXCF option of the IPCONFIG configuration statement is different than was previously specified. This indicates that the IP address for existing devices created by XCF dynamics should be changed. The failure occurred because an XCF device was still active. All devices generated by XCF dynamics must be stopped before the IP address change is performed.

System Action: TCP/IP continues.

User or Operator Response: Issue Display TCPIP,procname,NETSTAT,DEV to get a list of all of the currently defined devices. The names of devices generated by XCF Dynamics can either be IUTSAMEH, IUTIQDIO or the cpname of another VTAM® in the sysplex. Once all of the names have been determined, issue VARY TCPIP,,STOP commands for each of the devices.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0626I D...OMPROUTE(,OSPF|RIP|RTTABLE)

Explanation: This message is the result of the HELP OMPROUTE command.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0627I D...OMPROUTE,OSPF, <(LIST,<ALL|AREAS|INTERFACES|NBMA|NEIGHBORS|VLINKS>) | LSA|AREASUM|EXTERNAL|DATABASE|DBSIZE|INTERFACE|NEIGHBOR| ROUTERS|STATISTICS>

Explanation: This message is the result of the HELP OSPF command.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0628I D...OMPROUTE,RIP, <(LIST,<ALL|INTERFACES|ACCEPTED>)|INTERFACE|FILTERS>

Explanation: This message is the result of the HELP RIP command.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0629I D...OMPROUTE,RTTABLE<,DEST=IP_ADDR>

Explanation: This message is the result of the HELP RTTABLE command.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0630I D...NETSTAT,CACHINFO

Explanation: This message is the result of the DISPLAY HELP CACHINFO command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0631I <GATE | ROUTE> LIST REFERENCES LINKS WITH NO HOME ADDRESSES

Explanation: The gate list entries in the GATEWAY statement or the route list entries in the BEGINROUTES block reference link names with no home addresses. Each routing definition provided by an invalid gate or route list entry is ignored. SEE PREVIOUS MESSAGES - there were error messages generated while processing your profile.

System Action: TCP/IP continues.

User or Operator Response: Assign a home address for each link referenced in the gate or route list using the HOME statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0632I MULTIPATH *type* SUPPORT IS ENABLED

Explanation: Multipath support is enabled, where *type* is either: PERCONNECTION, or PERPACKET. In general, multipath routing provides the routing distribution necessary to balance the network utilization of outbound packets. Multipath routing requires the definition of multiple equal-cost routes which are either defined statically or added dynamically by routing protocols. If MULTIPATH is specified without any subparameters, the default is PERCONNECTION.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0635I VIPA *ip address* IS NOT IN NET *net* WITH MASK *subnet mask*

Explanation: All of the VIPAs in a single VIPADYNAMIC VIPADEFINE list must be in the same net. The indicated ip address was not in the same net, so it was deleted from the list which was in error. Other addresses in the list are processed. The required net is identified by the network prefix *net* and the address mask *mask*.

System Action: None.

User or Operator Response: If the ipaddress was incorrectly specified, correct the error and try the command or activation again. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0636I D...NETSTAT,VIPADYN D...SYSPLEX,VIPADYN<,IPADDR=> <,MAX=><,DEBUG>

Explanation: This message is the result of the DISPLAY HELP VIPADYN command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0637I D...SYSPLEX,VIPADYN

Explanation: This message is the result of the DISPLAY HELP SYSPLEX command and shows the help available for the Sysplex command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0638I PERMANENT LOOPBACK ADDRESS 127.0.0.1 SPECIFIED ON LINE *lineno* CANNOT BE USED IN A *vipad* STATEMENT

Explanation: The permanent loopback IP address 127.0.0.1 that is specified on line *lineno* cannot be used on a *vipad* (VIPADEFINE, VIPABACKUP, VIPARANGE, VIPADELETE, or VIPADISTRIBUTE) statement.

System Action: TCPIP ignores the statement. TCP/IP continues.

User or Operator Response: Correct the statement. Rerun the profile.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0639I ATMENABLED PARAMETER IS OBSOLETE, OSAENABLED USED

Explanation: The ATMENABLED parm on the SACONFIG statement is obsolete. This usually indicates that an unmodified profile from a prior release is being used. The ATMENABLED parm was replaced by the OSAENABLED parm. This includes the original ATMENABLED support, and support for SNMP management information from OSA-Express Ethernet devices. See *z/OS Communications Server: IP Configuration Reference* for more information about OSAENABLED and OSADISABLED parameters.

System Action: Profile processing continues as if the new OSAENABLED parm had been specified.

User or Operator Response: If you want SNMP management information for ATM OSA-2 or OSA-Express Ethernet devices, no action is required.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0641I IP FORWARDING *type* SUPPORT IS ENABLED

Explanation: The status of IP Forwarding is displayed, where *type* is one of the following:

- FWDMULTIPATH
- PERPACKET
- NOFWDMULTIPATH

Forwarding can be specified on the IPCONFIG (DATAGRAMFWD) statement or the ASSORTEDPARMS statement.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0650I <IP ADDRESS | ADDRESS MASK> 0.0.0.0 INCORRECT ON LINE *lineno*

Explanation: An incorrect IP address or address mask of 0.0.0.0 was specified on line *lineno*

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Check the profile statement at line *lineno* and specify a valid ip address.

Source Data Set: EZACFPPT, EZACFPAR

Procedure Name: parseFile

EZZ0651I NUMBER OF MASK BITS SPECIFIED ON LINE *lineno* **IS INCORRECT**

Explanation: The number of mask bits specified was incorrect. This number must be in the range of 1–32.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Check the profile statement at line *lineno* and respecify a valid number of mask bits.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0652I D...NETSTAT,ACCESS,NETWORK

Explanation: This message is the result of the DISPLAY HELP ACCESS command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0653I D...NETSTAT,VIPADCFG

Explanation: This message is the result of the DISPLAY HELP VIPADCFG command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0654I D...STOR,MODULE=XMODID

Explanation: This message is the result of the DISPLAY HELP STOR command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0655I PORT *number protocol* **IS ALREADY RESERVED**

Explanation: The PORT specified is already reserved.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0656I THE '=' SIGN IS NOT ALLOWED FOR *default_type statement* **ENTRY**

Explanation: An equal sign (=) may not be specified on a ROUTE or GATEWAY statement corresponding to a DEFAULT or DEFAULT6 route entry.

default_type is the DEFAULT or DEFAULT6 keyword

statement is the ROUTE or GATEWAY statement

System Action: Profile processing continues. However, this route entry will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0657I ROUTE LIST ENTRY ON LINE *lineno* **FOR DESTINATION** *destaddr* **IS UNREACHABLE THROUGH INTERFACE** *intfaceaddr* **ON** *interface*

Explanation: The route list entry within the ROUTE block on the line specified is in error. The destination address is unreachable through the specified interface using any known routes.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFMS1

EZZ0658I PARTIAL IP ADDRESS *ip_addr* **NOT ALLOWED ON ROUTE ENTRY STATEMENT ON LINE** *lineno*

Explanation: The specified route list entry within the ROUTE block on the line specified is in error. A fully qualified IP address must be specified.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Correct the statement and rerun the profile. For more information about the statement, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZACFPAR

EZZ0659I D...NETSTAT,VCRT<,DETAIL><,IPADDR=|PORT=>

Explanation: This message is the result of the DISPLAY HELP VCRT command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0660I D...NETSTAT,VDPT<,DETAIL><,IPADDR=|PORT=>

Explanation: This message is the result of the DISPLAY HELP VDPT command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0661I *function* **FOR PROFILE** *dsn* **FAILED:** *rc/rsn*, *errormsg*

Explanation: TCP/IP was unable to open or read the profile data set *dsn* because of an error. This message can be issued during the following:
- initial profile processing (the first profile data set referenced during TCP/IP initialization)
- INCLUDE profile statement processing
- processing of the profile data set specified on a VARY TCPIP,,OBEYFILE command

function is the Language Environment® function that failed.

dsn is the name of the input Profile data set being processed.

rc/rsn is the decimal return code and hexadecimal reason code from the Language Environment function that failed.

errormsg describes the error.

System Action: If the error occurs during initial profile processing and no profile data set was found using the normal search order, then TCP/IP ends. Otherwise, TCP/IP continues but the current OBEYFILE or INCLUDE data set cannot be processed.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the *rc/rsn*. See the *z/OS UNIX System Services Messages and Codes* for *rc/rsn* explanations.

Source Data Set: EZACFPPR

Procedure Name: cffopen2

EZZ0662I OPEN OF INTERNAL HIPERSPACE MEMORY FILE FAILED: *errormsg*, *dspservrc/dspservrsn*

Explanation: TCP/IP was unable to open an internal hiperspace memory file to process the initial PROFILE or an OBEYFILE because of the information in *errormsg*, *dspservrc/dspservrsn*.

errormsg is the Language Environment message that explains the error.

dspservrc is the hexadecimal return code from the MVS DSPSERV function that failed.

dspservrsn is the hexadecimal reason code from the MVS DSPSERV function that failed.

System Action: If this error occurs, TCP/IP attempts to use a temporary HFS file for the PROFILE and OBEYFILE processing. Execution continues.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the error information. See the *z/OS Language Environment Debugging Guide* for *errormsg* explanations. See the *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for MVS DSPSERV return/reason code explanations. If the error cannot be corrected because of constraints in the customer's environment, this message can be ignored.

Source Data Set: EZACFPPR

Procedure Name: cffopen2

EZZ0663I OPEN OF INTERNAL TEMPORARY HFS FILE FAILED: *rc/rsn*, *errormsg*

Explanation: TCP/IP tried to open an internal temporary HFS file in either the directory specified by the TMPDIR environment variable or, if TMPDIR is not defined, in the /tmp directory, in order to process the initial PROFILE or an OBEYFILE. The open failed because of the information in *rc/rsn* and *errormsg*.

rc/rsn is the decimal return code and hexadecimal reason code from the Language Environment function that failed.

errormsg describes the error.

System Action: If the error occurs during initial PROFILE processing TCP/IP ends. Otherwise, TCP/IP continues but the current VARY TCPIP,,OBEYFILE command cannot be processed.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the *rc/rsn*. See the *z/OS UNIX System Services Messages and Codes* for *rc/rsn* explanations.

Source Data Set: EZACFPPR

Procedure Name: cffopen2

EZZ0664I *function* **FOR INTERNAL HIPERSPACE MEMORY FILE FAILED:** *errormsg*, *hspservrc/hspservrsn*

Explanation: TCP/IP tried to execute function *function* against an internal hiperspace memory file while processing the initial PROFILE or an OBEYFILE, but the function failed because of the information in *errormsg*, *hspservrc/hspservrsn*.

function is the Language Environment function that failed.

errormsg is the Language Environment message that explains the error.

hspservrc is the hexadecimal return code from the MVS HSPSERV function that failed.

hspservrsn is the hexadecimal reason code from the MVS HSPSERV function that failed.

System Action: If this error occurs, TCP/IP attempts to use a temporary HFS file for the PROFILE and OBEYFILE processing. Execution continues.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the error information. See the *z/OS Language Environment Debugging Guide* for *errormsg* explanations. See the *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG* for MVS HSPSERV return/reason code explanations.

Source Data Set: EZACFPPR

Procedure Name: cffopen2

EZZ0665I *function* **FOR INTERNAL TEMPORARY HFS FILE FAILED:** *rc/rsn*, *errormsg*

Explanation: TCP/IP tried to execute function *function* against an internal temporary HFS file while processing the initial PROFILE or an OBEYFILE, but the function failed because of the information in *rc/rsn* and *errormsg*. The temporary HFS file is located either in the directory specified by the TMPDIR environment variable or, if TMPDIR is not defined, in the /tmp directory.

function is the Language Environment function that failed.

rc/rsn is the decimal return code and hexadecimal reason code from the Language Environment function that failed.

errormsg describes the error.

System Action: If the error occurs during initial PROFILE processing TCP/IP ends. Otherwise, TCP/IP continues but the current VARY TCPIP,,OBEYFILE command cannot be processed.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the *rc/rsn*. See the *z/OS UNIX System Services Messages and Codes* for *rc/rsn* explanations.

Source Data Set: EZACFPPR

Procedure Name: cffopen2

EZZ0666I **OPEN OF INTERNAL COMMAND FILE FAILED:** *rc*, *errormsg*

Explanation: The TCP/IP command processor was unable to open an internal file used to process commands because of the error indicated by *rc*.

rc is the decimal return code from the Language Environment fopen() function.

errormsg describes the error.

System Action: TCP/IP continues, but no DISPLAY or VARY commands will be processed.

User or Operator Response: None.

System Programmer Response: See the *z/OS UNIX System Services Messages and Codes* for an explanation of the *rc* value. See the *z/OS Language Environment Debugging Guide* for *errormsg* explanations.

Source Data Set: EZACFMMN, EZACFC00

Procedure Name: main, parseCmds

EZZ0667I *function* **FOR INTERNAL COMMAND FILE FAILED:** *rc*, *errormsg*

Explanation: TCP/IP tried to execute function *function* against an internal command file but the function failed because of *rc*. This message can be issued during DISPLAY/VARY TCPIP command processing.

function is the Language Environment function that failed.

rc is the decimal return code from the Language Environment function.

errormsg describes the error.

System Action: TCP/IP continues but the current command cannot be processed.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by *rc*. See the *z/OS UNIX System Services Messages and Codes* for *rc* explanations. See the *z/OS Language Environment Debugging Guide* for *errormsg* explanations. If the error cannot be corrected, report the error to the IBM Software Support Center.

Source Data Set: EZACFC00

Procedure Name: parseCmds

EZZ0668I D...TELNET,PARMSGROUP, <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,ID=XPGROUPNM> <,MAX=XNN|*>

Explanation: This message is the result of the DISPLAY HELP PARMSGROUP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0669I D...TELNET,PARMSMAP <,PORT=ALL|XNUM|XNUM1..XNUM2|BASIC|SECURE> <,PROFILE=XPRFID|CURR|ACT|ALL> <,MAX=XNN|*>

Explanation: This message is the result of the DISPLAY HELP PARMSMAP command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0670I D...TELNET,INACTLUS <,MAX=XNN|*>

Explanation: This message is the result of the DISPLAY HELP INACTLUS command and shows the format of the command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0671I NOT ENOUGH SPACE IN TMPDIR OR /TMP HFS DIRECTORY FOR PROFILE/OBEYFILE PROCESSING

Explanation: While processing an initial PROFILE or an OBEYFILE, TCP/IP received an out-of-space error on a temporary internal HFS file used in the processing. The temporary HFS file is located either in the directory specified by the TMPDIR environment variable or, if TMPDIR is not defined, in the /tmp directory.

System Action: If the error occurs during initial PROFILE processing TCP/IP ends. Otherwise, TCP/IP continues but the current VARY TCPIP,,OBEYFILE command was not processed.

User or Operator Response: None.

System Programmer Response: Ensure there is enough space to process the initial PROFILE or OBEYFILE, in either the directory specified by the TMPDIR environment variable or, if TMPDIR is not defined, in the /tmp directory.

Source Data Set: EZACFPPR

Procedure Name: cffopen2

EZZ0672I THE SOCKET CALL *function* **FOR THE AUTOLOG TASK FAILED:** *rc/rsn* **-** *errmsg*

Explanation: The autolog task issued a SETIBMOPT, SOCKET or IOCTL function call that failed.

function is the Language Environment function that failed.

rc is the decimal return code from the Language Environment function that failed. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

rsn is the hexadecimal reason code from the Language Environment function that failed. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

errmsg is a message that explains the meaning of the return code.

System Action: The autolog task ends and cannot be restarted until TCPIP is restarted.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the *rc/rsn*.

Source Data Set: EZACFALG

Procedure Name: main, do_BEGINPROFILE

EZZ0673I AUTOLOG TASK: SHUTDOWN IN PROGRESS

Explanation: The TCPIP stack requested that the autolog task shutdown. This can occur when a STOP command is issued for the TCPIP stack or during ESTAE error recovery for the configuration component.

System Action: The autolog task ends and cannot be restarted until TCPIP is restarted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFALG

Procedure Name: main

EZZ0674I THE SOCKET CALL *function* **FOR THE CONFIG SOCKET** *socknum* **FAILED:** *rc/rsn* **-** *errmsg*

Explanation: The configuration component acts as a server establishing a listener to which the TN3270 Server connects to receive configuration information and operator commands. This message is displayed whenever any of the socket calls (SOCKET, IOCTL, BIND, LISTEN, GETSOCKNAME and ACCEPT) used to establish the connection fails. In addition, a Netstat Drop issued for the listener connection ID will cause the ACCEPT socket call to fail.

function is the Language Environment function that failed.

socknum is the socket descriptor number for the config socket.

rc is the decimal return code from the Language Environment function that failed. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

rsn is the hexadecimal reason code from the Language Environment function that failed. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

errmsg is a message that explains the meaning of the return code.

System Action: Usually, the listener is established during TCPIP initialization. If the configuration component fails to establish the listener during initialization, it will retry whenever a TCPIP operator command is issued. Once the listener is established, a subsequent failure cannot be corrected until TCPIP is restarted.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by the *rc* and *rsn*.

Source Data Set: EZACFTEL

Procedure Name: getConfigPort, handleConfigSocket

EZZ0675I THE SOCKET CALL *function* **FOR THE CLIENT CONFIG SOCKET** *socknum* **FOREIGN PORT** *portnum* **ADDR** *ipaddr* **FAILED:** *rc/rsn* **-** *errmsg*

Explanation: The configuration component acts as a server establishing a listener to which the TN3270 Server connects to receive configuration information and operator commands. When the Telnet Server connects to the listener, the configuration component accepts the connection, creating the client config socket. This message is displayed whenever any of the socket calls (SELECTEX, RECV and SEND) used to communicate with the Telnet Server fails. Some common reasons for errors in the socket calls are:

A foreign application, such as a probe, connected to the configuration component listener producing unexpected results.

The Telnet Server ended.

A Netstat Drop was issued for this connection.

function is the Language Environment function that failed.

socknum is the socket descriptor number for the client config socket.

portnum is the port number used by the foreign socket.

ipaddr is the IP address used by the foreign socket.

rc is the decimal return code from the Language Environment function that failed. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

rsn is the hexadecimal reason code from the Language Environment function that failed. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

errmsg is a message that explains the meaning of the return code.

System Action: If the error occurs during initial profile processing the configuration component will close the socket. When the Telnet Server detects that the connection was closed, it will establish a new connection.

User or Operator Response: None.

System Programmer Response: If possible, correct the error indicated by *rc* and *rsn*.

Source Data Set: EZACFTEL

Procedure Name: recvFromClient, sendToClient

EZZ0676I THE CLIENT CONFIG SOCKET *socknum* **WAS CLOSED**

Explanation: The configuration component acts as a server establishing a listener to which the TN3270 Server connects in order to receive configuration information and operator commands. When the Telnet Server connects to the listener, the configuration component accepts the connection, creating the client config socket. This message is displayed whenever the RECV socket call determines that the socket was closed. Some common reasons for this error are:

A foreign application, such as a probe, connected to the config component listener then closed the connection.

The Telnet Server ended.

A Netstat Drop was issued for this connection.

socknum is the socket descriptor number for the client config socket.

System Action: The configuration component closes the socket. If the Telnet Server failed and can recover, it will establish a new connection.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFTEL

Procedure Name: recvFromClient

EZZ0677I THE CLIENT CONFIG SOCKET *socknum* **TIMED OUT**

Explanation: The configuration component acts as a server establishing a listener to which the TN3270 Server connects in order to receive configuration information and operator commands. When the Telnet Server connects to the listener, the configuration component accepts the connection, creating the client config socket. After the configuration component has sent information to the Telnet Server, it expects to receive a confirmation message from Telnet. This message is displayed whenever the SELECTEX socket call has not received the confirmation within five minutes. Some common reasons for this error are:

A foreign application, such as a probe, connected to the configuration component listener but does not send data.

The Telnet Server ended.

A Netstat Drop was issued for this connection.

socknum is the socket descriptor number for the client config socket.

System Action: The configuration component closes the socket. If the Telnet Server failed and can recover, it will establish a new connection.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: recvFromClient

Procedure Name: EZACFTEL

EZZ0678I THE CLIENT CONFIG SOCKET RECEIVED AN INCORRECT HEADER PROTOCOL FROM TELNET: *hexvalue*

Explanation: The configuration component acts as a server establishing a listener to which the TN3270 Server connects to receive configuration information and operator commands. When the Telnet Server connects to the listener, the configuration component accepts the connection, creating the client config socket. After the configuration component has sent information to the Telnet Server, it expects to receive a confirmation message from Telnet. This message is displayed if the confirmation message contains incorrect information. The most likely reason for this error is that a foreign application, such as a probe, has connected to the configuration component listener and has sent incorrect data.

hexvalue is the hexidecimal display of the header protocol that was received.

System Action: The configuration component closes the socket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFTEL

Procedure Name: recvFromClient

EZZ0679I THE CLIENT CONFIG SOCKET *socknum1* **IS ALREADY IN CONTACT AS TELNET SOCKET** *socknum2*

Explanation: The configuration component acts as a server establishing a listener to which the TN3270 Server connects to receive configuration information and operator commands. When the Telnet Server connects to the listener, the configuration component accepts the connection, creating the client config socket and saving the socket information in the Telnet socket field. This message is displayed whenever unsolicited data is received on a new connection while there still is information in the Telnet socket field. The most likely reason for this error is that a foreign application, such as a probe, has connected to the configuration component listener and has sent data that matches Telnet header specifics.

socknum1 is the current socket descriptor number for the client config socket.

socknum2 is the previous socket descriptor number for the client config socket.

System Action: The configuration component sends a message to the new socket with the 'Already in Contact' indicator turned on and closes the new socket.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFTEL

Procedure Name: handleClientSocket

EZZ0683I GLOBALCONFIG *parm* **CONTAINS AN INCORRECT VALUE** *value*

Explanation: A parameter on the GLOBALCONFIG profile statement contains an incorrect value.

parm is the parameter for which an incorrect value was specified.

value is the specified incorrect value. Two possible reasons that *value* is incorrect are:

- The value is outside of the acceptable range.
- The value is within the acceptable range, but would cause an immediate storage shortage because the amount of storage currently allocated is greater than or equal to 80% of the specified value.

System Action: Profile processing continues. However, the parameter value is not defined properly and will be ignored.

User or Operator Response: None.

System Programmer Response: Correct the statement and rerun the profile. For more information about the GLOBALCONFIG statement, see the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0684I CANNOT ENABLE DVIPSEC SUPPORT AFTER TCPIP IS ACTIVE

Explanation: DVIPSEC is an option that can be specified following IPCONFIG FIREWALL. This option is only valid on an initial profile. This message will be displayed if IPCONFIG FIREWALL DVIPSEC is coded in an obeyfile.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: If DVIPSEC function is needed, then bring TCP/IP down. Code IPCONFIG FIREWALL DVIPSEC in your initial profile and start TCP/IP again.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0685I D...NETSTAT,STATS<,PROTOCOL=><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP STATS command and shows the format of the command.

System Action: TCPIP continues.

User or Operator Response: Refer to the*z/OS Communications Server: IP System Administrator's Commands* for more information.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0686I *parameter* **VALUE IN BEGINROUTE BLOCK ON LINE** *lineno* **CONFLICTS WITH PREVIOUS ROUTE DEFINITIONS**

Explanation: While processing a BEGINROUTES block that starts on line number *lineno*, a route definition was encountered that conflicts with a previous route definition.

parameter is the conflicting parameter, and its values are:

REPLACEABLE
A replaceable route was defined to a destination to which a NOREPLACEABLE route has already been defined in this BEGINROUTES block. A mixture of replaceable and non-replaceable routes cannot be defined to the same destination.

NOREPLACEABLE
A non-replaceable route was defined to a destination to which a replaceable route has already been defined in this BEGINROUTES block. A mixture of replaceable and non-replaceable routes cannot be defined to the same destination.

lineno is the line number at which the BEGINROUTES block starts.

System Action: The route definition is rejected and processing continues with the next route defined in the BEGINROUTES block.

User or Operator Response: Contact the system programmer

System Programmer Response: Correct the statement and rerun the profile. For more information about the BEGINROUTES block, see the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: EZACFMS1

Procedure Name: ioctlmessage

EZZ0687I **ROUTE LIST ENTRY** *entrynum* **ON LINE** *linenum* **FOR DESTINATION** *destaddr* **CONFLICTS WITH SYSPLEX DISTRIBUTOR TARGET DVIPA**

Explanation: Replaceable static routes cannot be defined to an address that corresponds to a dynamic VIPA for which this stack is a sysplex distributor target.

entrynum is the BEGINROUTES block entry that is in error.

linenum is the line on which the BEGINROUTES block begins

destaddr is the destination IP address for which the replaceable static route definition failed.

System Action: The incorrect route definition is rejected. Processing continues with the next route defined in the BEGINROUTES block.

User or Operator Response: Correct the route definition.

System Programmer Response: Correct the route definition.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0688I **IQDIO ROUTING IS** *status*

Explanation: This message shows the status of IQDIO ROUTING.

status is one of the following:

ENABLED
The IPCONFIG IQDIOROUTING parm was processed successfully and iQDIO routing is in effect.

DISABLED
iQDIO routing is not in effect. Either the IPCONFIG NOIQDIOROUTING parm was specified, or an OBEYFILE with the IPCONFIG NODATAGRAMFWD parameter was processed. iQDIO routing cannot be performed without datagram forwarding.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0689I CANNOT ENABLE IQDIO ROUTING - *reason*

Explanation: iQDIO routing cannot be enabled.

reason is one of the following:

Firewall is active
iQDIO routing cannot be enabled if firewall support is enabled (IPCONFIG FIREWALL).

IP Forwarding is disabled
iQDIO routing cannot be enabled if IP forwarding is disabled (IPCONFIG NODATAGRAMFWD).

Processor is not Hipersocket capable
iQDIO routing cannot be enabled if the processor on which TCP/IP is running is not capable of sending data using Hipersockets.

TCPIP activated with NOIQDIOROUTING
iQDIO routing cannot be enabled if IQDIOROUTING was not specified at TCP/IP initialization.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: getData

EZZ0690I D...NETSTAT,IDS<,PROTOCOL=|SUMMARY>

Explanation: This message is the result of the DISPLAY HELP IDS command and shows the format of the command.

System Action: TCPIP continues.

User or Operator Response: See to the *z/OS Communications Server: IP System Administrator's Commands* for more information.

System Programmer Response: None.

Source Data Set: EZACFHLP

Procedure Name: parseFile

EZZ0691I DEVICE NAME *devname* **ON LINE** *linenum* **DOES NOT HAVE A VALID CHPID**

Explanation: The device name contains a chpid that is not valid. The device name must be configured using the reserved prefix IUTIQDxx, where the suffix 'xx' must be equal to the hexadecimal value of the desired IQD CHPID (00 - FF).

devname is the device name that is incorrect.

linenum is the line number where the error occurred.

System Action: TCP/IP continues.

User or Operator Response: Correct the device name, specifing a valid chpid.

System Programmer Response: Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the MPCIPA Device.

Source Data Set: EZACFPAR

Procedure Name: parsefile

EZZ0692I CLASS D IP ADDRESS *ipaddr* **NOT ALLOWED ON LINE** *linenum*

Explanation: A class D IP address was specified on a HOME list entry on line *linenum*. This is not allowed.

ipaddr is the IP address that was specified.

linenum is the line number where the IP address was specified.

System Action: TCP/IP continues, but not all entries on the HOME list are processed.

User or Operator Response: None.

System Programmer Response: Check the profile statement on line *linenum* and specify a valid IP address that is not class D.

Source Data Set: Parsefile

Procedure Name: EZACFPAR

EZZ0695I *configstmt* **STATEMENT ON LINE** *lineno* **NOT VALID - IPV6 SUPPORT IS NOT ENABLED**

Explanation: This message warns the user that a configuration statement is not valid because the stack is not enabled for IPv6.

configstmt is one of the following:

- PKTTRACE
- INTERFACE
- IPCONFIG6
- PORT

lineno is the line number in the profile/obeyfile where the error is found.

System Action: TCP/IP continues but the statement in error is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0696I *stmt_or_cmd* **NOT VALID - IPV6 SUPPORT IS NOT ENABLED**

Explanation: A configuration statement or command is invalid because the stack is not enabled for IPv6.

stmt_or_cmd is one of the following:

- PKTTRACE command
- IPv6 ROUTE statement
- DATTRACE command
- IPCONFIG6 statement
- INTERFACE statement
- PORT statement

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0697I FORMAT SHORT IGNORED - IPV6 SUPPORT IS ENABLED

Explanation: The IPCONFIG FORMAT keyword only applies to stacks that are not enabled for IPv6. If the stack is enabled for IPv6, then the FORMAT keyword has no meaning.

System Action: Processing continues. FORMAT LONG is in effect.

User or Operator Response: None.

System Programmer Response: Correct the IPCONFIG FORMAT keyword in your profile/obeyfile.

Source Data Set: EZACFPAR

Procedure Name: Parsefile

EZZ0699I IPV6 FORWARDING IS DISABLED

Explanation: The status of IPv6 forwarding is displayed. IPv6 forwarding can be disabled by specifying NODATAGRAMFWD on the IPCONFIG6 statement.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0700I IPV6 FORWARDING *type* SUPPORT IS ENABLED

Explanation: The status of IPv6 Forwarding is displayed. IPv6 Forwarding can be enabled by specifying DATAGRAMFWD on the IPCONFIG6 statement.

type is either FWDMULTIPATH PERPACKET or NOFWDMULTIPATH.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0701I ICMPV6 REDIRECTS <will|will not> BE IGNORED

Explanation: The state of ICMPv6 redirects is displayed.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0702I IPV6 SOURCEVIPA SUPPORT IS *status*

Explanation: The status of IPv6 SourceVipa supports is displayed.

status is either ENABLED or DISABLED

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0703I IPV6 MULTIPATH SUPPORT IS DISABLED

Explanation: The status of IPv6 Multipath support is displayed. If Multipath support is disabled and there are multiple equal-cost paths to a destination, then TCP/IP will use the first path found for all IPv6 packets to that destination.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0704I IPV6 MULTIPATH *type* SUPPORT IS ENABLED

Explanation: IPv6 Multipath support is enabled. In general, multipath routing provides the routing distribution necessary to balance the network utilization of outbound packets. Multipath routing requires the definition of multiple equal-cost routes which are either defined statically or added dynamically by routing protocols. If IPCONFIG6 MULTIPATH is specified without any subparameters, the default is PERCONNECTION.

type is either PERCONNECTION or PERPACKET.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0705I D...NETSTAT,ND<,IPADDR=><,FORMAT=LONG|SHORT>

Explanation: This message is the result of the DISPLAY HELP ND command and shows the format of the command.

System Action: TCPIP continues.

User or Operator Response: Refer to the *z/OS Communications Server: IP System Administrator's Commands* for more information about the DISPLAY HELP command.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0706I TCPSTACKSOURCEVIPA IS IGNORED - SOURCEVIPA IS NOT ENABLED

Explanation: The TCPSTACKSOURCEVIPA parameter was specified on an IPCONFIG Profile statement but SOURCEVIPA is not enabled. SOURCEVIPA must be enabled for the TCPSTACKSOURCEVIPA function to be enabled.

System Action: TCP/IP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: If you want TCPSTACKSOURCEVIPA support then respecify the IPCONFIG statement with the SOURCEVIPA parameter.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0707I *keyword1* **AND** *keyword2* **ARE MUTUALLY EXCLUSIVE ON SAME** *configstmt* **STATEMENT - STATEMENT IS IGNORED**

Explanation: The keywords specified on the configstmt statement are mutually exclusive. The statement is ignored.

keyword1 and *keyword2* are the IP Configuration keywords specified on the configstmt statement.

configstmt is the IP Configuration statement where the error was found.

System Action: TCP/IP continues, but the statement in error is ignored.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the configuration statement in error.

Source Data Set: EZACFYAC

Procedure Name: errorCheck

EZZ0708I **PERMANENT LOOPBACK ADDRESS ::1 ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: The permanent loopback address **::1** cannot be specified on the IP Configuration statement at line *lineno*.

lineno is the line number in the profile or obeyfile where the error was found.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the statement. Rerun the profile.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0709I **LINK LOCAL ADDRESS OF** *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: A link local address cannot be specified on the IP Configuration statement at line *lineno*.

ipv6addr is the IPv6 address.

lineno is the line number in the profile or obeyfile where error was found.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the statement. Rerun the profile.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0710I **MULTICAST ADDRESS OF** *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: A multicast address cannot be specified on the IP Configuration statement at line lineno.

ipv6addr is the IPv6 address.

lineno is the line number in the profile or obeyfile where error was found.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the statement. Rerun the profile.

Source Data Set: EZACFPV6

Procedure Name: ProcessInterface

EZZ0711I IP ADDRESS *ipv6addr* **NO LONGER DEPRECATED**

Explanation: The IP address specified on the INTERFACE statement is no longer deprecated. This will happen when the user codes an INTERFACE ADDADDR *ipv6addr* for a previously deprecated address.

ipv6addr is the IPv6 address.

Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the INTERFACE statement.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0713I IPV4-MAPPED IPV6 ADDRESS OF of *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: An IPv4-mapped IPv6 address cannot be specified on the IP configuration statement at line lineno.

ipv6addr is the IPv6 address.

lineno is the line number in the profile or obeyfile where the error was found.

Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the IP configuration statement where the error was found.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0714I IPV4-COMPATIBLE IPV6 ADDRESS OF *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: An IPv4-compatible IPv6 address cannot be specified on the IP configuration statement at line lineno.

ipv6addr is the IPv6 address.

lineno is the line number in the profile or obeyfile where the error was found.

Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the IP configuration statement where the error was found.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0715I IPV6 ADDRESS ON LINE *lineno* **IS NOT VALID**

Explanation: An invalid IPv6 address was specified on the IP configuration statement at line lineno.

lineno is the line number in the profile or obeyfile where the error was found.

Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the IP configuration statement where the error was found.

System Action: TCP/IP ignored the statement.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFV6

Procedure Name: processInterface

EZZ0716I *parm* **PARAMETER ON LINE** *lineno* **IS OBSOLETE**

Explanation: The parameter specified is obsolete.

parm is the IP configuration parameter that is obsolete.

lineno is the line number in the profile or obeyfile where the parameter was found.

Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the IP configuration statement where the error was found.

System Action: TCP/IP continues, but the configuration parameter is ignored.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFPPT

Procedure Name: processPort

EZZ0717I *configstmt* **STATEMENT ON LINE** *lineno* **WILL BE RETIRED IN A FUTURE RELEASE**

Explanation: The IP configuration statement indicated by *configstmt* will be retired in a future release. Refer to the description of *configstmt* in the *z/OS Communications Server: IP Configuration Reference* for more information about other configuration statements that can be used in place of *configstmt*.

configstmt is the name of the configuration statement.

lineno is the line number in the profile or obeyfile where the statement was found.

System Action: The statement is processed and TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFPAR

Procedure Name: parseFile

EZZ0718I **INTERFACE** *interfacename* **ON LINE** *lineno* **IS** *interface_state*

Explanation: An attempt to change the state of the interface failed because of the reason stated.

interfacename is the IPv6 interface name.

lineno is the line number in the profile or obeyfile where the statement was found.

interface_state is the reason for the failure and is one of the following:

ALREADY STARTED
An attempt was made to start an interface that was already started.

ALREADY STOPPED
An attempt was made to stop an interface that was already stopped.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlmsg

EZZ0719I ROUTER ADVERTISEMENT HOP LIMIT VALUES WILL BE IGNORED

Explanation: Router advertisement hop limit values will be ignored. This is a result of coding IGNOREROUTERHOPLIMIT on the IPCONFIG6 statement. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the IPCONFIG6 statement.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: EZACFMS1

Procedure Name: ioctlMsg

EZZ0720I ROUTER ADVERTISEMENT HOP LIMIT VALUES WILL NOT BE IGNORED

Explanation: Router advertisement hop limit values will not be ignored. This is a result of coding NOIGNOREROUTERHOPLIMIT on the IPCONFIG6 statement. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the IPCONFIG6 statement.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACFMS1

Procedure Name: ioctlMsg

EZZ0721I ON LINE *linenum* **AN ATTEMPT WAS MADE TO DEFINE MORE INTERFACES FOR** *type name* **THAN IS ALLOWED**

Explanation: The port name specified has more interfaces already defined than is supported.

linenum is the line number in the profile or obeyfile where the error was found.

type is the PORTNAME parameter on the INTERFACE statement. PORTNAME is used to specify the PORT name contained in the TRLE definition for the QDIO interface.

name is the port name that is specified on the INTERFACE statement for OSA-Express Gigabit Ethernet and QDIO Fast Ethernet.

System Action: Profile processing continues. However, the statement is not defined properly and will be ignored.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement. See the *z/OS Communications Server: IP Configuration Reference* for more information about the statement.

Source Data Set: EZACFMS1

Procedure Name: parseFile

EZZ0722I V...PURGECACHE,XNAME

Explanation: This message is the result of the D TCPIP,,HELP,PURGECACHE command.

System Action: TCP/IP continues.

User or Operator Response: For more information about the command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDHLP

Procedure Name: parseFile

EZZ0723I CANNOT DELETE ALL IP ADDRESSES ON INTERFACE *intf_name*

Explanation: You cannot use INTERFACE DELADDR to delete all IP addresses that are defined to a virtual interface.

intf_name is the interface name on the INTERFACE statement where DELADDR is being attempted.

System Action: TCP/IP continues, but all IP addresses that were specified on INTERFACE DELADDR for *intf_name* have not been deleted.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFMS1

Procedure Name: ioctlMsg

EZZ0724I IPv6 ADDRESS OF *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: An unspecified IPv6 address cannot be specified on the IP configuration statement at line *lineno*.

ipv6addr is the unspecified IPv6 address.

lineno is the line number in the profile or obeyfile where the address is not allowed.

System Action: TCP/IP ignores the statement. TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information about the statement in error.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0726I RESERVED IPv6 ADDRESS OF *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: A reserved IPv6 address cannot be specified on the IP configuration statement at line lineno.

A reserved IPv6 address is one of the following:

- Universal/local bit is set (bit 71).
- Individual/group bit is set (bit 72).
- The first four bytes of the interface ID are X'00005EFE'.
- The first 57 bits of interface ID, except universal/local and individual/group bits, are set to 1 (for example, FCFFFFFFFFFFFF8/57).

Refer to the *z/OS Communications Server: IPv6 Network and Application Design Guide* for information about IPv6 addresses.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0727I RESERVED SITE-LOCAL IPv6 ADDRESS OF *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: A reserved site-local address cannot be specified on the IP configuration statement at line *lineno*.

Refer to the *z/OS Communications Server: IPv6 Network and Application Design Guide* for information about the format of a site-local IPv6 address.

ipv6addr is the reserved IPv6 site-local address.

lineno is the line number in the profile or obeyfile where the address is not allowed.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFPV6

Procedure Name: processInterface

EZZ0728I RESERVED SITE-LOCAL IPv6 PREFIX OF *ipv6addr* **ON LINE** *lineno* **CANNOT BE SPECIFIED**

Explanation: A reserved site-local prefix cannot be specified on the IP configuration statement at line *lineno*.

Refer to the *z/OS Communications Server: IPv6 Network and Application Design Guide* for information about the format of a site-local IPv6 address.

ipv6addr is the reserved IPv6 site-local prefix.

lineno is the line number in the profile or obeyfile where the prefix is not allowed.

System Action: TCP/IP ignored the statement. TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the statement and use the VARY TCPIP,,OBEYFILE command to rerun the configuration statement.

Source Data Set: EZACFPV6

Procedure Name: processInterface

Chapter 2. EZZ2*xxxx* messages

EZZ2351I Incorrect option: *option***.**

Explanation: You specified an incorrect option.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Correct the syntax of the incorrect option, and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDNET

Procedure Name: parsONet or parsTNet

EZZ2352I Missing parameter after option *option***.**

Explanation: You specified an *option* option without a value.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Specify a parameter after *option*, and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: parsONet, parsTNet

EZZ2353I Extraneous parameter *parameter***.**

Explanation: You specified an extraneous parameter.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter, and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: parsONet, parsTNet

EZZ2354I Incorrect integer parameter *parameter* **or the integer is over its maximum value**

Explanation: You specified a parameter which is not numeric when a numeric parameter was expected, or the value is over its maximum.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter, and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: tokValCh, optArgCh, filtVchk

© Copyright IBM Corp. 1994, 2002

EZZ2355I *argument* **specified for** *parameter* **must be** *Length* **characters or less.**

Explanation: You specified an incorrect *parameter* argument.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Specify an *argument* less than or equal to 8 characters and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: filtVchk, keyValCh

EZZ2356I **More than one** *option* **was specified.**

Explanation: You specified more than one *option* option.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Remove extraneous *option* and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: parsONet or parsTNet

EZZ2357I **You specified more than the maximum of** *number* **select values.**

Explanation: You specified more than the maximum of *number* select values.

number is the maximum number of select values allowed. The maximum is either **1** for the DEvlinks/-d select option, or **6** for the rest of Netstat select options

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Remove extraneous select values and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: filtVchk, keyValCh

EZZ2359I **Select options can be done only with** *options***.**

Explanation: Select options are not supported on the onetstat/NETSTAT option you requested.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Specify select options with a correct option and resubmit the onetstat/NETSTAT command.

The correct *options* for NETSTAT are ALL, ALLConn, BYTEinfo, CLients, COnn, DEvlinks, Gate, ND, ROUTe, TELnet, VCRT, VDPT: REPort, and STACk are valid output options.

The correct *options* for onetstat are -A, -a, -b, -c, -d, -e, -g, -n, -r, -t, -O and -V options.

For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: parsONet or parsTNet

EZZ2360I Select option *option* **can be done only with** *option***.**

Explanation: This SELECT option is not supported on the onetstat/NETSTAT option you requested.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Specify select options with a correct option and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: parsONet or parsTNet

EZZ2364I Incorrect prefix length *prefixLen*

Explanation: You specified an incorrect prefix length. For an IPv4 address, the prefix length range is 1–32 and for an IPv6 address, the prefix length range is 1–128.

prefixLen is the prefix length.

System Action: The Netstat program halts and exits.

User or Operator Response: Specify a correct prefix length and reissue the Netstat command. For information about the Netstat command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, and EZACDTNE

Procedure Name: parsONET() and parsTNET()

EZZ2365I Incorrect IP address *ipaddress***.**

Explanation: You specified an incorrect IP address. This message will also be issued if a wildcard character is specified for the options that do not support it.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Specify a correct IP address and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE, EZACDDNE, EZACDNEQ, EZACDNE1, EZACDNE2, EZACDNE6

Procedure Name: parsONet, parsTNet, cdDne, procGate, setIpAddr, setIpAddr6, procND

EZZ2366I Incorrect subnet mask *subnetMask***.**

Explanation: You specified an incorrect subnet mask.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Specify a correct subnet mask and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDNE1

Procedure Name: setIpAddr

EZZ2367I Parenthesis but no search criteria

Explanation: You specified a parenthesis but no search criteria.

System Action: The NETSTAT program halts and exits.

User or Operator Response: Specify search criteria or remove the parenthesis and resubmit the NETSTAT command. For information about the NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTNE

Procedure Name: parsTNET

EZZ2369I Onetstat is a UNIX shell command and must be issued from the UNIX shell.

Explanation: The onetstat command was not issued from a UNIX shell as required.

System Action: The onetstat program halts and exits.

User or Operator Response: Reissue the command from a UNIX shell. For information about the onetstat, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE

Procedure Name: main

EZZ2370I *ioctl* **failed with error :** *error* **(***errno***/***errnojr***).**

Explanation: The onetstat/NETSTAT issued the SIOCGIBMOPT ioctl for the *ioctl* subtype. The ioctl call failed with the specified error.

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

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Correct the indicated error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the indicated error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDNE0, EZACDNE1, EZACDNE2, EZACDNE6

Procedure Name: getIcmpInfo, getIpInfo, getTcpInfo, getUdpInfo, getDatCfg, getGlbCfg, getIpCfg, getTcpCfg, getUdpCfg, getHomeList, getIntfTab, getIfData, getPktTab, getTcpProcName, getTcpCnnTable, getUdpListen, getMvsMibStruct, getPostList, getAutoLogList, getRtTable, getIcmpInfo6, getIpInfo6, getIpCfg6, getTcpCnnTable6, getUdpListen6, getRtTable6, procARP, procDROP, procCACH, procVIPA, procSLAP, procACCN, procDVCF, procVCRT, procVDPT, procIDS, procDROP6, procND

EZZ2371I *devicename***, specified for link** *linkname***, was not found.**

Explanation: The onetstat/Netstat command processor was parsing the TCP/IP interface table to get the -d option response information. A link record was found which had a link name of *linkname* and a device name of *devicename*, but the TCP/IP interface table did not contain a device record for that device.

System Action: The onetstat/NETSTAT program continues.

User or Operator Response: This problem could occur if a device or link was added to TCPIP while the onetstat/NETSTAT was processing the interface table. Retry the onetstat/NETSTAT request. If the problem persists,

check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact the IBM Software Support Center.

System Programmer Response: Check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact the IBM Software Support Center.

Source Data Set: EZACDNE0, EZACDNE6

Procedure Name: procDEVL, procDEV6

EZZ2372I Incorrect connection number *connId***.**

Explanation: You specified an incorrect connection number.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: To find the correct connection number, issue onetstat -c. Specify a correct connection number and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDNE0, EZACDNE6

Procedure Name: procDROP, procDEV6

EZZ2373I NETSTAT: DROP connection process failed RACF authorization checking

Explanation: You attempted to drop a connection, but your user ID does not have MVS.VARY.TCPIP.DROP defined in the RACF profile.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Make sure that your user ID has the MVS.VARY.TCPIP.DROP defined in the RACF profile and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDNE1

Procedure Name: netAuthCheck

EZZ2374I Unable to open message catalog *catalogfilename* **:** *error*

Explanation: Netstat was unable to open the message catalog *catalogfilename* in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be "NLSPATH=/usr/lib/nls/msg/%L/%N".

System Action: Netstat will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: main

EZZ2375I sigaction() failed for *signal* **:** *reason*

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

Function
Description

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

SIGTERM
handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, certain resources may not be properly cleaned up when a SIGTERM is received.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: main

EZZ2376I Could not determine the TCPIPjobname, using default of 'INET'

Explanation: Onetstat called the z/OS UNIX service __iptcpn() to retrieve the resolver supplied TCPIPjobname and failed. Onetstat could not determine the jobname for the TCP/IP stack. A default value of 'INET' will be used for TCPIPjobname.

System Action: The onetstat program continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for onetstat to communicate with a particular stack, the TCPIPjobname should be set in the appropriate resolver configuration file or data set. Make sure that the TCPIPjobname statement in the appropriate resolver configuration file or data set is correct and resubmit the onetstat command. For more information about the search order for locating the resolver configuration file or data set, see the *z/OS Communications Server: IP Configuration Guide*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The onetstat -p option can be used to explicitly select a TCP/IP instance by specifying its jobname.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: main

EZZ2377I Could not establish affinity with *tcpipname* (*error_code*/*reason*) - can not provide the requested option information

Explanation: Onetstat called setibmopt() to associate itself with the TCPIP instance *tcpipname*, and failed with the displayed *error_code* and *reason*. The requested option information cannot be provided.

System Action: The onetstat program halts and exits.

User or Operator Response: Correct the error indicated by *error_code* and *reason* and reissue the onetstat command. For onetstat to communicate with a particular stack, the *tcpipname* (as determined by system variable TCPIPjobname) must match the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTYPE statement that defined in the BPXPRMxx parmlib member. For more information about the customizing the BPXPRMxx parmlib member, see *z/OS UNIX System Services Planning*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The onetstat -p option can be used to explicitly select a TCP/IP instance by specifying its jobname.

System Programmer Response: Correct the error indicated by *error_code* and *reason* and reissue the onetstat command. For onetstat to communicate with a particular stack, the *tcpipname* (as determined by system variable TCPIPjobname) must match the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTYPE statement that defined in the BPXPRMxx parmlib member. For more information about the customizing the BPXPRMxx parmlib member, see *z/OS UNIX System Services Planning*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The onetstat -p option can be used to explicitly select a TCP/IP instance by specifying its jobname.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: main

EZZ2378I Termination request is received. The program is interrupted.

Explanation: This message is displayed when a termination request is received while the onetstat command is in progress.

System Action: The onetstat program halts and exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: termHand

EZZ2379I Command abend with *abendcode*, *reasoncode*.

Explanation: This message is displayed when an internal programming error caused onetstat program to be abnormally terminated.

System Action: The onetstat program halts and exits.

User or Operator Response: If possible, recreate the problem with -z option to collect debug information and forward the results to the TCPIP administrator.

System Programmer Response: The abend information will be written to the syslog output. Contact the IBM Software Support Center with the syslog output.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: abndHand

EZZ2380I Could not open file *filename* **for writing (***error_code***/***reason***).**

Explanation: NETSTAT called fopen() to open the file *filename* and failed with the displayed *error_code* and *reason*. The requested option information cannot be provided.

System Action: The NETSTAT program halts and exits.

User or Operator Response: Correct the error indicated by *error_code* and *reason* and reissue the NETSTAT command. For information about the fopen(), see *z/OS C/C++ Programming Guide*.

System Programmer Response: Correct the error indicated by *error_code* and *reason* and reissue the NETSTAT command. For information about the fopen(), *z/OS C/C++ Programming Guide*.

Source Data Set: EZACDTNE

Procedure Name: parsTNet

EZZ2381I Incorrect option argument: *argument*.

Explanation: You specified an incorrect option argument.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Correct the syntax of the incorrect option argument, and resubmit the onetstat/NETSTAT command. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: parsONet or parsTNet

EZZ2382I Unable to open *type* **socket to** *tcpipname***:** *error*

Explanation: The onetstat/NETSTAT must open a UDP or RAW socket to TCP/IP in order to retrieve the requested information from TCP/IP. The onetstat/NETSTAT's attempt to open the socket failed due to the specified error.

System Action: The onetstat/NETSTAT program halts and exits.

User or Operator Response: Correct the problem indicated by *error*. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the problem indicated by *error*. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDONE, EZACDTNE

Procedure Name: openSock

EZZ2383I Cannot provide the SHORT format Netstat *option* **report for the IPv6 enabled stack** *tcpipname*

Explanation: The user requested the SHORT format Netstat *option* report for the stack *tcpipname*. Because the tcpipname stack is IPv6 enabled, a SHORT format report is not allowed.

option is the Netstat option.

tcpipname is the TCP/IP stack name.

System Action: The Netstat program halts and exits.

User or Operator Response: Reissue the Netstat command without the FORMAT option or with the FORMAT LONG option. For information about the Netstat command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDDNE, EZACDONE, and EZACDTNE

Procedure Name: main()

EZZ2384I Missing close quotation for DSN name.

Explanation: You specified a fully qualified DSN name without a close quotation.

System Action: The NETSTAT program halts and exits.

User or Operator Response: Correct the syntax of the incorrect DSN name, and resubmit the NETSTAT command. For information about the NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTNE

Procedure Name: parsTNet

EZZ2385I Access to Netstat *option* **denied - SAF RC is** *safrc*

Explanation: The user attempted to execute Netstat *option*, but the user ID is not authorized to the Netstat security product profile for *option*.

option is the Netstat option.

safrc is the hexadecimal SAF return code.

System Action: The Netstat program halts and exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: Locate the specified return code in your installed host security product's documentation to determine the cause of the error. If you are using the z/OS Security Server (RACF) as your security product, the SAF return code is documented in the *z/OS Security Server RACROUTE Macro Reference* in the section about return codes for the RACROUTE REQUEST=AUTH function. Your installed host security product might have

issued messages regarding the failure. See the *z/OS Communications Server: IP System Administrator's Commands* for more information about the Netstat security product resource name definitions.

Source Data Set: EZACDNE1

Procedure Name: netAuthCheck

EZZ2386I Failure in Netstat authorization processing - Function code is *functioncode* **RC is** *rc*

Explanation: The Netstat authorization process failed.

functioncode is the function code.

rc is the return code.

System Action: The Netstat program halts and exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: If TCP/IP CTRACE was active when the problem occurred, CTRACE records should be written at the time of the error. Otherwise, recreate the problem with CTRACE active for option IOCTL. Contact the IBM Support Center with the CTRACE records and message information.

Source Data Set: EZACDNE1

Procedure Name: netAuthCheck, netIoctl

EZZ2387I TCP/IP procname is not active

Explanation: The Netstat command attempted to retrieve information from the TCP/IP stack procname, but this stack is not currently active.

procname is the TCP/IP name.

System Action: The Netstat program halts and exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: Start the TCP/IP stack named procname, and reissue Netstat command.

Source Data Set: EZACDNE1

Procedure Name: netAuthCheck, netIoctl

EZZ2388I *function* **failed due to error** *h_errno*

Explanation: The function *function* issued by the **onetstat** command failed with error *h_errno*. Possible causes of the problem are an error in specifying the configuration file for the Resolver, or that the Resolver address space was not started.

function is the C/C++ Run-Time function that failed.

h_errno is one of the following errors:

1 HOST_NOT_FOUND

2 TRY_AGAIN

3 NO_RECOVERY

4 NO_DATA

System Action: For some function failures, the **onetstat** command ends. For other function failures, the **onetstat** command continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the Resolver address space is started. See the *z/OS Communications Server: IP Configuration Guide* for information about starting the Resolver address space. If this does not resolve the problem, correct the error indicated by the h_errno value. See the *z/OS C/C++ Run-Time Library Reference* for a description of the *h_errno* for the function that failed.

Source Data Set: EZACDONE

Procedure Name: main

EZZ2389I Unable to provide *reqInfo* information because *tcpipname* is not IPv6 enabled

Explanation: The user requested the *reqInfo* information for the stack *tcpipname*. Because the *tcpipname* stack is not IPv6 enabled, the requested information is not available.

reqInfo is the requested information.

tcpipname is the TCP/IP stack name.

System Action: The Netstat program halts and exits.

User or Operator Response: The requested information is only available on an IPv6 enabled stack. If you need the requested information, reissue the Netstat command against an IPv6 enabled stack. For information about the onetstat/NETSTAT command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDDNE, EZACDONE, and EZACDTNE

Procedure Name: main()

EZZ2500I NETSTAT versionRelease

Explanation: This message displays the current version and release for the command. The message is followed by the output for the requested command report. For a detailed description of the report, see the TSO NETSTAT section of the *z/OS Communications Server: IP System Administrator's Commands*.

System Action: The Display Netstat command continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDNE0, EZACDNE6

Procedure Name: procACNN(), procALLC(), procARP(), procBYTE(), procCNFG(), procCONN(), procDEVL(), procDVCF(), procHOME(), procIDS(), procPORT(), procROUT(), procSOCK(), procSTAT(), procVCRT(), procVDPT(), procVIPA(), procALLC6(), procBYTE6(), procCNFG6(), procCONN6(), procDEVL6(), procHOME6(), procND6(), procPORT6(), procROUT6(), procSOCK6(), procSTAT6()

Chapter 3. EZZ3*xxxx* messages

EZZ3105I *function* **failed due to error** *h_errno*

Explanation: The function *function* issued by the **oping** command failed with error *h_errno*. Possible causes of the problem are an error in specifying the configuration file for the Resolver, or that the Resolver address space was not started.

function is the C/C++ Run-Time function that failed.

h_errno is one of the following errors:

1 HOST_NOT_FOUND

2 TRY_AGAIN

3 NO_RECOVERY

4 NO_DATA

System Action: For some function failures, the **oping** command terminates. For other function failures, the **oping** command continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the Resolver address space is started. See the *z/OS Communications Server: IP Configuration Guide* for information about starting the Resolver address space. If this does not resolve the problem, correct the error indicated by the h_errno value. See the *z/OS C/C++ Run-Time Library Reference* for a description of the *h_errno* for the function that failed.

Source Data Set: EZACDPIN

Procedure Name: pingMain

EZZ3107I Oping is a UNIX shell command and must be issued from a UNIX shell.

Explanation: The oping command was not issued from a UNIX shell, as required.

System Action: The oping program halts and exits.

User or Operator Response: Reissue the command from a UNIX shell. For information about the oping, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDOPN

Procedure Name: main

EZZ3108I Unable to open message catalog ″pingmsg.cat″ : *error*

Explanation: Ping was unable to open the message catalog ″pingmsg.cat″ in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be ″NLSPATH=/usr/lib/nls/msg/%L/%N″.

System Action: Ping will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: main

© Copyright IBM Corp. 1994, 2002

EZZ3109I Extraneous parameter *parameter***.**

Explanation: You specified an extraneous parameter.

System Action: The Ping program halts and exits.

User or Operator Response: Correct the syntax of the incorrect parameter and resubmit the Ping command. For information about the Ping command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: parsOPIN, parsTPIN

EZZ3110I Unknown option *option***.**

Explanation: You specified an unknown option.

System Action: The Ping program halts and exits.

User or Operator Response: Check the unknown option for misspellings or other problems. Correct the option and resubmit the Ping command. For information about the Ping command, see *z/OS Communications Server: IP System Administrator's Commands*. Also consult related MAIN pages and the online help, if available.

System Programmer Response: None.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: parsOPIN, parsTPIN

EZZ3111I Unknown host *name***.**

Explanation: The host name specified could not be resolved to an IP address.

System Action: The Ping program halts and exits.

User or Operator Response:

1. Correct the syntax of the host name and resubmit the Ping command. For information about the Ping command, see *z/OS Communications Server: IP System Administrator's Commands*.
2. Check that the specified host name is valid. If the host name looks correct, check with your system programmer to verify the host address.
3. Use the IP address, if it is known.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: procHost

EZZ3112I Host name or address not entered.

Explanation: The Ping command was issued with no host name or IP address requested.

System Action: The Ping program halts and exits.

User or Operator Response: Reissue the Ping command with the host identification included.

System Programmer Response: None.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: parsOPIN, parsTPIN

EZZ3113I Missing value after *option* **option.**

Explanation: You specified an *option* option without a value.

System Action: The Ping program halts and exits.

User or Operator Response: Specify an *option* value between the accepted minimum and maximum values and resubmit the Ping command. For information about the Ping command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: parsOPIN, parsTPIN, tokValChk

EZZ3114I The value of *option* **must be between** *minvalue* **and** *maxvalue***.**

Explanation: You specified an incorrect *option* option value.

System Action: The Ping program halts and exits.

User or Operator Response: Specify an *option* value between the accepted minimum and maximum values and resubmit the PING command. For information about the Ping command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: optValChk, tokValChk

EZZ3115I Unable to open RAW socket: *error*

Explanation: The Ping attempted to open a raw socket in order to issue an ICMP Echo request for the Ping function. The Ping was unable to open the raw socket for the reason specified by *error*.

System Action: The Ping program halts and exits.

User or Operator Response: Correct the error indicated. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the error indicated. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDPIN

Procedure Name: openSock

EZZ3116I sigaction() failed for *signal* **:** *reason*

Explanation: The Ping program encountered an error attempting to set up the signal handler for the signal specified by *signal*.

reason is the error returned by the C run-time library for the failing sigaction() call. If the signal handler is not correctly enabled, the Ping program will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

Function
Description

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

SIGTERM
handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, certain resources may not be properly cleaned up when a SIGTERM is received.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDOPN, EZACDTPN

Procedure Name: main

EZZ3117I Termination request is received. The program is interrupted.

Explanation: This message is displayed when a termination request is received while the Ping command is in progress.

System Action: The Ping program halts and exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: termHand

EZZ3118I Command abend with *abendcode*, *reasoncode*.

Explanation: This message is displayed when an internal programming error caused Ping program to be abnormally terminated.

System Action: The Ping program halts and exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center with the CEEDUMP output from your system.

Source Data Set: EZACDPIN

Procedure Name: abndHand

EZZ3119I Interaction attention request is received. The program is interrupted.

Explanation: This message is displayed when an interaction attention request is received while the Ping command is in progress.

System Action: The Ping program is interrupted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: intHand

EZZ3120I Could not establish affinity with *tcpipname* **(***error_code***/***reason***)**

Explanation: Ping called setibmopt() to associate itself with the TCPIP instance *tcpipname*, and failed with the displayed *error_code* and *reason*.

System Action: The Ping program halts and exits.

User or Operator Response: Correct the error indicated by *error_code* and *reason* and reissue the Ping command. For Ping to communicate with a particular stack, the *tcpipname* (as determined by system variable TCPIPjobname) must match the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTYPE statement that was defined in the BPXPRMxx parmlib member. For more information about customizing the BPXPRMxx parmlib member, see *z/OS UNIX System Services Planning*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their job names.

System Programmer Response: Correct the error indicated by *error_code* and *reason* and reissue the Ping command. For Ping to communicate with a particular stack, the *tcpipname* (as determined by system variable TCPIPjobname) must match the TYPE operand that was specified on the FILESYSTYPE statement or the NAME operand of the SUBFILESYSTYPE statement that was defined in the BPXPRMxx parmlib member. For more information about customizing the BPXPRMxx parmlib member, see *z/OS UNIX System Services Planning*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their job names.

Source Data Set: EZACDPIN

Procedure Name: pingMain

EZZ3121I Executing under single stack configuration of INET. The *option* option ignored

Explanation: The Ping command was issued from an environmental shell that is configured for INET. In an INET configuration, there can be only one TCP/IP (for example, AF_INET type) stack connected to an environmental shell. The Ping program continues but ignores the *option* option.

option is the Ping option that was ignored.

System Action: The Ping program continued.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: pingMain

EZZ3122I *argument* specified for *parameter* must be *Length* characters or less.

Explanation: You specified an incorrect *parameter* argument.

System Action: The Ping program halts and exits.

User or Operator Response: Specify an *argument* less than or equal to requested characters and resubmit the Ping command. For information about the Ping command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDPIN, EZACDOPN, EZACETPN

Procedure Name: procINTF, parsOPIN, parsTPIN

EZZ3123I Could not determine the TCPIPjobname, using default of 'INET'

Explanation: The oping command was invoked with the -i option, but without the -p option to specify a TCP/IP instance name. The command called the __iptcpn() to retrieve the resolver supplied TCPIPjobname and failed. Oping could not determine the jobname for the TCPIP stack. A default value of 'INET' will be used for TCPIPjobname.

System Action: The oping program continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for oping to communicate with a particular stack, either the TCPIPjobname should be set in the appropriate resolver configuration file or data set, or the command should be reissued, supplying the TCP/IP instance name on the -p option. For more information about the search order for locating the resolver configuration file or data set, see the *z/OS Communications Server: IP Configuration Guide*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their job names. The oping -p option can be used to explicitly select a TCP/IP instance by specifying its jobname.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: pingMain

EZZ3125I The interface *interface* was not found in the HOME list for *tcpname*.

Explanation: The interface *interface* was specified on the INTF/-i option but was not defined to the TCP/IP instance named *tcpname*. If the TCP/IP instance name is 'TCPIP' this could mean that Ping is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used.

System Action: The Ping program halts and exits.

User or Operator Response: Verify that the value specified for the INTF/-i option is defined to the TCP/IP instance named *tcpname*. You can use the onetstat or Netstat command to verify the interfaces defined to a TCP/IP instance. Reissue the Ping command with a valid value for the INTF/-i option. For more information about Ping, see the *z/OS Communications Server: IP System Administrator's Commands*,

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: getIFindx

EZZ3126I Interface *interface* for *tcpname* can not be specified for the *option* option.

Explanation: The Ping command was invoked with the *option* option to specify the interface on which to send out the Ping packets. The interface *interface* was found in the HOME list of the TCP/IP instance name *tcpname*, but the interface type cannot be specified for the *option* option.

If the TCP/IP instance name is 'TCPIP' this could mean that Ping is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used.

option is the Ping option that was specified.

System Action: The Ping program halts and exits.

User or Operator Response: Ensure that the interface specified for the *option* option is not a VIPA link or loopback link and reissue the command.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: getIFindx

EZZ3127I More than one IP address *ipaddr* found for *tcpname*.

Explanation: The Ping command was invoked with an IP address specified for the INTF/-i option but there is more than one interface defined to the TCP/IP instance *tcpname* with IP address *ipaddr*. If the TCP/IP instance name is 'TCPIP' this could mean that Ping is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used.

System Action: The Ping program halts and exits.

User or Operator Response: Reissue the Ping command with a valid value for the INTF/-i option.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: getIFindx

EZZ3129I IPv4-mapped IPv6 address not supported

Explanation: IPv4-mapped IPv6 addresses are not supported for the Ping command.

System Action: The Ping program halts and exits.

User or Operator Response: Correct the IP address specification and reissue the Ping command. For information about the Ping command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: procHOST, procSRIP, procINTF

EZZ3130I Incorrect *option* value specified

Explanation: You specified an incorrect *option* value.

option is the command option for which the incorrect value was specified.

System Action: The Ping program halts and exits.

User or Operator Response: Correct the *option* value and reissue the Ping command. For information about the Ping command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: procSRIP, procINTF, getIFindx

EZZ3131I Found parenthesis but no options

Explanation: You specified a parenthesis without specifying any options.

System Action: The Ping program halts and exits.

User or Operator Response: Specify options or remove the parenthesis and reissue the Ping command. For information about the Ping command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTPN

Procedure Name: parsTPIN

EZZ3132I Unable to retrieve HOME list for *tcpname* - *description* (*return_code* / *reason_code*)

Explanation: The command was unable to obtain the HOME list from TCP/IP stack *tcpname*. If the TCP/IP stack name is TCPIP this might mean that the command is executing in an INET environment and is not aware of the actual name of the TCP/IP stack being used.

tcpname is the name of the TCP/IP stack from which Ping tried to obtain the HOME list.

description describes the meaning of the return code.

return_code is the decimal UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

reason_code is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The Ping program halts and exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: If the *return_code* and *reason_code* were set by the TCP/IP stack, re-create the problem with CTRACE options RAW, PFS, and IOCTL active. Contact the IBM Support Center with the CTRACE. If the *return_code* and *reason_code* were not set by the TCP/IP stack, contact the product that set the *return_code* and *reason_code* for assistance.

Source Data Set: EZACDPIN

Procedure Name: getIFindx

EZZ3133I Mismatched IP address type values

Explanation: The command found a mismatch in IP address type (that is, IPv4 or IPv6) between some of the IP address values specified. IP address type can be specified with the ADDRTYPE/-A option; or by providing an IP address as the destination host, SRCIP/-s option value, or INTF/-i option value. If a LINK or INTERFACE name was

specified for the INTF/-i option value, the command will verify that the LINK or INTERFACE is of the same IP address type as the host or other IP address options specified.

System Action: The Ping program halts and exits.

User or Operator Response: Verify that the values specified for destination host, and the ADDRTYPE/-A, SRCIP/-s, or INTF/-i options, are all of the same IP address type, IPv4 or IPv6.

System Programmer Response: None.

Source Data Set: EZACDPIN

Procedure Name: pingMain, procHOST, procSRIP, procINTF

EZZ3200I SNMP subagent: Internal Error *code*

Explanation: The subagent encountered an internal programming error.

code is one of the following:

01 mkDPIopen failed

02 failure parsing dpi packet (DPIopen)

03 no DPI response to DPI open

04 agent rejected the Open request - reason unknown

05 subagent not authorized to agent

06 DPIget_fd_for_handle failed

07 failure during mkDPIregister

08 failure parsing DPI packet (DPIregister)

09 Sever error processing packet

0A subagent could not get TSEB address

0B subagent could not get TSDB address

0C subagent could not get TSDX address

0D subagent control block pointer was NULL

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: If possible, re-create the problem with subagent trace level 4. Contact the TCPIP administrator.

System Programmer Response: Error information will be written to the Syslog Daemon (syslogd) output file. Contact your IBM support center with the syslogd output.

Source Data Set: EZASADPI.C

Procedure Name: do_connect_and_open, do_register, call_function, main

---

EZZ3201I SNMP subagent: duplicate subagent identifier error

Explanation: The SNMP Agent rejected the DPI open request from the subagent because another subagent has already connected to the Agent using the same subagent identifier.

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: The subagent identifier for the MVS SNMP subagent is 1.3.6.1.4.1.2.11.7.2. Ensure that no user DPI programs are using this subagent identifier.

System Programmer Response: The subagent identifier for the MVS SNMP subagent is 1.3.6.1.4.1.2.11.7.2. Ensure that no user DPI programs are using this subagent identifier.

Source Data Set: EZASADPI.C

Procedure Name: do_connect_and_open

EZZ3202I SNMP SUBAGENT: INITIALIZATION COMPLETE

Explanation: The SNMP subagent completed initialization and is ready to start processing requests.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3203I SNMP SUBAGENT: RESTART SCHEDULED

Explanation: An attempt was made to automatically restart the SNMP subagent following a severe error which caused the SNMP subagent to be ended. This message is preceded by an error message indicating why the SNMP subagent was ended.

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: exitsuba

EZZ3204I SNMP subagent: sigaction() failed for *signal* **:** *reason*

Explanation: The SNMP subagent encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, the subagent will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

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 may not be properly cleaned up. The subagent may not be automatically restarted.

SIGTERM handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, the subagent will not be automatically restarted when a SIGTERM is received.

SIGPIPE handler allows the subagent to detect when the connection to the SNMP agent was terminated by the agent (for example, if the agent times out while waiting for the subagent response and closes the connection). If sigaction fails for SIGPIPE, subagent may either hang or terminate without restart.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3205I SNMP SUBAGENT: SHUTDOWN IN PROGRESS

Explanation: The SNMP subagent is permanently ending. This could be caused by a STOP command being issued for the TCPIP address space. The subagent is automatically started at TCP/IP initialization unless SACONFIG DISABLED is specified in the TCP/IP Profile data set. So when the TCP/IP address space is stopped, the subagent issues this message and then ends. SNMP subagent ending will also occur if the SNMP subagent encounters an error too severe to attempt automatic restart.

System Action: The SNMP subagent is ended.

User or Operator Response: If the message is issued due to a STOP command, no action is necessary. If this

message occurs following an unrecoverable subagent error, it will be preceded by one or more error messages indicating the error that caused the subagent to be ended. Correct the errors listed by the previous error messages.

System Programmer Response: If the message is issued due to a STOP command, no action is necessary. If this message occurs following an unrecoverable subagent error, it will be preceded by one or more error messages indicating the error that caused the subagent to be ended. Correct the errors listed by the previous error messages. TCP/IP must be ended and restarted to restart the subagent.

Source Data Set: EZASADPI.C

Procedure Name: main, endsuba, exitsuba, abndhand

EZZ3206I SNMP subagent: unable to open UDP socket to TCPIP: *error*

Explanation: The SNMP subagent must open a UDP socket to TCP/IP in order to retrieve the SNMP variable information from TCP/IP. The subagent's attempt to open the UDP socket failed due to the specified error.

System Action: An automatic restart of the subagent is attempted.

User or Operator Response: Correct the problem indicated by *error*. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the problem indicated by *error*. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3207I SNMP subagent: possible restart loop detected

Explanation: An error occurred in the SNMP subagent that caused a restart to be attempted. The restart processing determined that the subagent had already been restarted multiple times within a short time span and therefore ended the restart processing to prevent a restart loop from occurring.

System Action: The SNMP subagent is ended. In order to restart the SNMP subagent, TCP/IP must be stopped and restarted. The MIB variables provided by the subagent will be unavailable until TCP/IP is restarted.

User or Operator Response: This message will be preceded by several error messages which will indicate the errors that occurred in the SNMP subagent to cause the subagent to be restarted. Correct the errors indicated by the previous error messages, then stop and restart TCPIP to restart the SNMP subagent.

System Programmer Response: This message will be preceded by several error messages which will indicate the errors that occurred in the SNMP subagent to cause the subagent to be restarted. Correct the errors indicated by the previous error messages, then stop and restart TCP/IP to restart the SNMP subagent.

Source Data Set: EZASADPI.C

Procedure Name: main, exitsuba, abndhand

EZZ3208I SNMP subagent: *ioctl* **failed with error:** *error*

Explanation: The subagent issued the SIOCGIBMOPT ioctl for the *ioctl* subtype. The ioctl call failed with the specified error.

System Action: The subagent is unable to process the request for the MIB variable. An error response is returned to the Agent.

User or Operator Response: Correct the indicated error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the indicated error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZASAIOC.C

Procedure Name: getIPLtime, sd_get_icmp_tbl, sd_get_ifEntry_tbl, sd_get_ip_tbl, sd_get_ipAddrEntry_tbl, sd_get_ipForwardEntry_tbl, sd_get_ipNetToMediaEntry_tbl, sd_get_tcp_tbl, sd_get_tcpConnEntry_tbl, sd_get_udp_tbl, sd_get_udpEntry_tbl, sd_get_netman_tbls, sd_get_mvsSys_tbl, sd_get_mvsPortEntry_tbl, sd_set_ip_tbl,

sd_set_mvs_tbl, sd_set_tcp_tbl, sd_set_udp_tbl, sd_set_primaryInterface, sd_set_ifAdminStatus, sd_drop_tcpConn, sd_drop_udpConn, lookup_linkName, do_beginProfile, do_endProfile

---

EZZ3209I SNMP subagent: device *device_name***, specified for link** *link_name***, was not found**

Explanation: The SNMP subagent was parsing the TCP/IP interfaces table to build the SNMP interfaces MIB variable table. A link record was found which had a linkname of *link_name* and a device name of *device_name*, but the TCP/IP interfaces table did not contain a device record for that device.

System Action: The subagent continues processing the interfaces table, but a stack table entry will not be made for the link in error.

User or Operator Response: This problem could occur if a device or link was added to TCPIP while the subagent was processing the interfaces table. Retry the SNMP request. If the problem persists, contact the TCPIP administrator.

System Programmer Response: Check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact your IBM Software Support Center.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_ifEntry_tbl

EZZ3210I SNMP subagent: interface index *index* **not found in SNMP interfaces table**

Explanation: The SNMP subagent was parsing the TCP/IP interfaces data table to add the interface counters to the SNMP interfaces MIB records built from processing the TCP/IP interfaces table. An interface data record was found that did not have a corresponding interface record.

System Action: The subagent continues processing the interfaces data table, but the SNMP interfaces MIB table will not contain data for the listed link.

User or Operator Response: This problem could occur if a device or link was added to TCPIP while the subagent was processing the interfaces table. Retry the SNMP request. If the problem persists, contact the TCP/IP Administrator.

System Programmer Response: Check the PROFILE.TCPIP file and ensure that the device definition for the listed device is specified in the file correctly. If the PROFILE.TCPIP file appears to be correct contact your IBM Software Support Center.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_ifEntry_tbl

EZZ3211I SNMP subagent: unable to open ping port: *error*

Explanation: The subagent attempted to open a raw socket in order to issue an ICMP Echo request for the remote ping function. The subagent was unable to open the raw socket for the reason specified by *error*.

System Action: The subagent will not be able to perform the remote ping request and will return an error for the remote ping MIB variable.

User or Operator Response: Correct the error indicated. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the error indicated. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZASARPG.C

Procedure Name: mkPINGport

EZZ3212I SNMP subagent: unable to open message catalog ″subamsg.cat″ : *error*

Explanation: The subagent attempted to open the subagent message catalog ″subamsg.cat″ in the message catalog directory, but was unable to open the catalog. The subagent message catalog should have been installed in the /usr/lib/nls/msg/C message catalog directory.

System Action: The subagent will use the internal default messages instead of the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3213I SNMP subagent: waiting for group *mib_tree*

Explanation: The SNMP subagent attempted to register the specified MIB tree, *mib_tree*, but another DPI subagent had already registered the requested MIB tree with a higher priority, or the SNMP subagent had successfully registered the specified MIB tree, but the Agent received a later registration from another subagent which requested a higher priority than that with which the SNMP subagent was currently registered.

System Action: The subagent will continue processing for the other MIB trees supported by the subagent. If the failing MIB tree becomes available at a later time the subagent will begin processing for that MIB tree.

User or Operator Response: If it is acceptable that a DPI subagent program other then the MVS SNMP subagent is providing the processing for the MIB variables in the specified MIB tree, than no action is necessary. Otherwise, the other DPI subagent must be ended in order for the MVS SNMP subagent to provide the processing for the variables in the specified MIB tree.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: do_register, do_unreg

EZZ3214I SNMP subagent: group *mib_tree* **unregistered by Manager**

Explanation: The SNMP subagent received a notification that the MIB tree *mib_tree* was unregistered. This action was initiated by a request from an SNMP Manager.

System Action: The subagent will continue processing for the other MIB trees supported by the subagent. The variables under the MIB tree unregistered by the SNMP Manager will no longer be available.

User or Operator Response: If it is acceptable that the specified MIB tree is no longer available, then no action is necessary. To regain the unregistered MIB tree, the SNMP subagent must be closed by an SNMP Manager. Do this by setting the saStatus for the SNMP subagent to invalid (2). This will cause the SNMP subagent to disconnect from the Agent, then reconnect and reregister all of its supported MIB trees, including any which were previously unregistered by an SNMP Manager request.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: do_unreg

EZZ3215I SNMP SUBAGENT: COULD NOT ESTABLISH AFFINITY WITH *tcpip_name* **(***error_code***/***reason***)**

Explanation: The SNMP subagent attempted to use the z/OS UNIX socket call, setibmopt(), to associate itself with the TCP/IP instance *tcpip_name*. This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifer' format of the MVS Start command was used) of the TCP/IP instance under which the SNMP subagent is initializing. The setibmopt call failed with the displayed *error_code* and *reason*.

System Action: The SNMP subagent terminates.

User or Operator Response: Most likely, the TCP/IP instance's name was not defined correctly to OMVS. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure OMVS. Ensure that the TCP/IP started procedure name (or identifier if the 'S member.identifier' format of the MVS Start command was used) matches the SUBFILESYSTYPE NAME. Recycle OMVS or TCP/IP if a change is necessary. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their names. If none of the above error conditions exist contact your system programmer.

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

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3216I SNMP SUBAGENT: LOST CONNECTION TO SNMP AGENT

Explanation: The SNMP subagent was connected to the SNMP Agent, but the connection was broken.

System Action: The subagent will try to reconnect to the SNMP Agent until successfully reconnected.

User or Operator Response: If the SNMP Agent job is not active, restart the SNMP Agent. If the SNMP Agent is currently active, the SNMP subagent should automatically reconnect to the agent. If it does not, stop the SNMP Agent and restart it.

System Programmer Response: If the SNMP Agent job is not active, restart the SNMP Agent. If the SNMP Agent is currently active, the SNMP subagent should automatically reconnect to the agent. If it does not, stop the SNMP Agent and restart it.

Source Data Set: EZASADPI.C

Procedure Name: do_open_and_register

EZZ3217I SNMP SUBAGENT: RECONNECTED TO SNMP AGENT

Explanation: The SNMP subagent reconnected to the SNMP Agent after detecting that the prior connection had been broken.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: do_open_and_register

EZZ3218I SNMP SUBAGENT: CONNECTED TO OSA/SF

Explanation: The SNMP subagent connected to the OSA/SF program.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAATM.C

Procedure Name: initAtm

EZZ3219I SNMP SUBAGENT: DISCONNECTED FROM OSA/SF

Explanation: The SNMP subagent had been connected to the OSA/SF program, but detected an error while attempting to communicate with OSA/SF. The OSA/SF connection is ended.

System Action: The subagent will try to recontact the OSA/SF program

User or Operator Response: Ensure that the OSA/SF program and ATM device are active. If possible, re-create the problem with subagent trace level 4. Contact the TCPIP administrator.

System Programmer Response: The trace messages should indicate the error which caused the subagent to disconnect from the OSA/SF program. If possible, correct the indicated error. If necessary, contact the IBM Software Support Center with the syslogd output.

Source Data Set: EZASAATM.C

Procedure Name: termAtm

EZZ3220I SNMP subagent: Using loopback to connect to agent

Explanation: The TCP/IP subagent was unable to resolve the local host address and is using the loopback address to connect to the snmp agent instead of the host address.

The TCP/IP subagent uses the gethostid() socket function to retrieve the local host address. The IP address returned by this function is the primary interface address of the TCP/IP stack associated with the subagent. If the returned IP address is loopback, or the gethostid() function fails, then the subagent will use loopback to connect to the SNMP agent and will issue this message.

System Action: The subagent will try to connect to the agent using the loopback address.

User or Operator Response: Contact the TCP/IP administrator.

System Programmer Response: If the TCP/IP subagent should not use loopback to connect to the agent, ensure that there is a non-loopback IP address defined as the primary interface to the TCP/IP stack associated with the subagent. The primary interface is either the first LINK in the HOME list, or the LINK specified on a PRIMARYINTERFACE Profile statement. You can use the TSO NETSTAT HOME or OS/390 UNIX onetstat/netstat -h commands to determine which LINK is the primary interface for a stack.

If the loopback address is used to connect to the agent, and a password other than the snmp agent's ″-c″ default password is used by the subagent when connecting, then the password used by the subagent must be defined for the loopback address 127.0.0.1 in the SNMP agent's PW.SRC or SNMPD.CONF file. The trace messages will indicate the error that the subagent received when attempting to retrieve the primary interface IP address. If necessary, contact your IBM Software Support Center with the syslogd output.

Source Data Set: EZASADPI.C

Procedure Name: do_gethostid

EZZ3221I SNMP SUBAGENT: SET REQUESTS *status*

Explanation: This message displays the current status of SNMP SET requests, where *status* is either: ENABLED or DISABLED. If the *status* is ENABLED, the SNMP subagent will accept SNMP SET requests. If the *status* is DISABLED, the SNMP subagent will reject SNMP SET requests.

This value can be set by specifying the desired parameter on the SACONFIG Profile statement. See the *z/OS Communications Server: IP Configuration Reference* for a description of the SACONFIG statement. This message will be issued during SNMP subagent initialization and whenever the value is changed dynamically using the VARY TCPIP,,OBEYFILE command.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASADPI.C

Procedure Name: main

EZZ3222I SNMP Subagent: no Netman response for *linkname* **(***seqnum***)**

Explanation: When an SNMP command is issued for an object from one of the ibm3172 MIB tables, and this table's cachetime expired, the subagent attempts to retrieve the information for a 3172 device or link from the 3172 ICP. The subagent then waits 30 seconds for a response from the 3172. If this response is not received within 30 seconds, the subagent writes this message to the syslog daemon indicating the *linkname* and the sequence number, *seqnum*, for the failing request.

System Action: The subagent fails the request for the object from the ibm3172 MIB table and waits for the next request.

User or Operator Response: If possible, re-create the problem with subagent trace level 3 and TCP/IP CTRACE active for options LCS, VTAM and VTAMDATA. Contact the TCPIP administrator.

System Programmer Response: Verify that the 3172 indicated by *linkname* is active. The subagent trace will be written to the Syslog Daemon (syslogd) output file. See the *z/OS Communications Server: IP Diagnosis* for information about obtaining the output of the CTRACE. Contact your IBM support center with the syslogd and CTRACE output.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_ibm3172_tbls

EZZ3223I SNMP Subagent: Netman request failed for *linkname* **rc** *netman_rc* **(***seqnum***)**

Explanation: When an SNMP command is issued for an object from one of the ibm3172 MIB tables, and this table's cachetime expired, the subagent attempts to retrieve the information for a 3172 device or link from the 3172 ICP. If the return code from the 3172 ICP is not zero, the subagent writes this message to the syslog daemon indicating the *linkname*, the *netman_rc*, and the sequence number, *seqnum*, of the failing request.

System Action: The subagent fails the request for the object from the ibm3172 MIB table and waits for the next request.

User or Operator Response: If possible, re-create the problem with subagent trace level 3 and TCP/IP CTRACE active for options LCS, VTAM and VTAMDATA. Contact the TCPIP administrator.

System Programmer Response: Verify that the 3172 indicated by *linkname* is active. The subagent trace will be written to the Syslog Daemon (syslogd) output file. See the *z/OS Communications Server: IP Diagnosis* for information about obtaining the output of the CTRACE. Contact your IBM support center with the syslogd and CTRACE output.

Source Data Set: EZASAIOC.C

Procedure Name: sd_get_ibm3172_tbls

EZZ3224I SNMP SUBAGENT: IP FORWARDING IS DISABLED

Explanation: IP forwarding support is disabled. The state of IP forwarding support was changed as the result of an SNMP SET request on either the ipForwarding or the ibmMvsIpForwarding MIB objects. MIB object.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3225I SNMP SUBAGENT: ICMP *(WILL | WILL NOT)* **IGNORE REDIRECTS**

Explanation: The state of ICMP redirects is displayed. The state was changed as the result of an SNMP SET request on the ibmMvsIgnoreRedirect MIB object.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3226I SNMP SUBAGENT: *(A LIMIT | NO LIMIT)* **ON INCOMING UDP DATAGRAM QUEUE SET**

Explanation: The limit on the incoming UDP datagram queue is displayed. The state was changed as the result of an SNMP SET request on the ibmMvsNoUdpQueueLimit MIB object.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3227I SNMP SUBAGENT: MULTIPATH SUPPORT IS DISABLED

Explanation: The state of MULTIPATH support was changed as the result of an SNMP SET request on the ibmMvsMultipathType MIB object.

If there are multiple equal-cost paths to a destination, then TCP/IP will use the first path found for all IP packets to that destination.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3228I SNMP SUBAGENT: PATH MTU DISCOVERY SUPPORT IS *status*

Explanation: The status of Path MTU Discovery support is displayed, where *status* is either: ENABLED or DISABLED. If Path MTU Discovery support is enabled, then TCP/IP will dynamically discover the Path MTU(PMTU), which is the minimum of the MTUs of each hop in the path, necessary to prevent fragmentation of datagrams sent on that path.

If Path MTU Discovery support is disabled, then large datagrams may be fragmented.

The state of Path MTU Discovery support was changed as the result of an SNMP SET request on the ibmMvsPathMtuDscEnabled MIB object.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3229I SNMP SUBAGENT: MULTIPATH *type* **SUPPORT IS ENABLED**

Explanation: Multipath support is enabled, where *type* is either: PERCONNECTION, or PERPACKET. In general, multipath routing provides the routing distribution necessary to balance the network utilization of outbound packets. Multipath routing requires the definition of multiple equal-cost routes which are either defined statically or added dynamically by routing protocols. If MULTIPATH is specified without any subparameters, the default is PERCONNECTION.

The state of MULTIPATH support was changed as the result of an SNMP SET request on the ibmMvsMultipathType MIB object.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3230I SNMP SUBAGENT: IP FORWARDING *type* **SUPPORT IS ENABLED**

Explanation: The state of IP forwarding support was changed as the result of an SNMP SET request on either the ipForwarding or ibmMvsIpForwarding MIB object. IP forwarding support is enabled, where *type* is either: NOFWDMULTIPATH, or FWDMULTIPATH PERPACKET.

System Action: The subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASAIOC.C

Procedure Name: do_endProfile

EZZ3250I THE MODIFY COMMAND IS NOT SUPPORTED.

Explanation: The TCPIP address space no longer supports the MODIFY command.

System Action: TCPIP continues.

User or Operator Response: A VARY operator command may be issued instead. See the *z/OS Communications Server: IP System Administrator's Commands* for assistance.

System Programmer Response: None.

EZZ3251I *job name* **IS NOT ACCEPTING VARY OR DISPLAY COMMANDS AT THIS TIME.**

Explanation: The internal state of this TCPIP instance indicates it cannot currently accept VARY or DISPLAY commands. Either TCPIP is not fully initialized, the Configuration component abnormally terminated, or TCPIP is terminating. The command request cannot be completed.

job name is the name of job associated with the procedure that was used to start TCPIP.

System Action: TCPIP continues.

User or Operator Response: Reissue the command. If the problem persists, save the system log and notify the system programmer.

System Programmer Response: Either TCPIP was in the process of terminating or there is a problem with the configuration component. The configuration component may either not be initialized or has abended and could not recover. If configuration cannot initialize, TCPIP will not start. Examine the system log to determine whether or not the configuration component abended and whether or not it was related to a correctable system configuration error. See the *z/OS Communications Server: IP Diagnosis* for assistance.

EZZ3252I ERROR *returncode* **IN ALLOCATING STORAGE FOR CONTROL BLOCK** *cbname*

Explanation: An error occurred while attempting to allocate storage for the control block specified. The return code from the STORAGE OBTAIN is provided.

returncode is the return code from the storage allocation routine. See the *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for a description of the STORAGE OBTAIN return codes.

cbname is the name of the control block for which the storage could not be obtained. The possible names are:

- SATQ - SNMP Subagent TrapQ block
- CDMQ - VARY/DISPLAY Command Request block
- SATQHEAD - SNMP Subagent TrapQ Header block
- SANQHEAD - SNMP Subagent NetmQ Header block

System Action: If the storage was for a CDMQ control block, TCPIP continues but the command process is terminated. If the storage was for a TRAPQ control block, the trap will not be generated.

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

System Programmer Response: The storage allocation attempt was for storage in common. Expanding the CSA may alleviate the problem. See the *z/OS MVS Initialization and Tuning Guide* for allocating more CSA.

EZZ3253I ERROR *returncode* **IN RELEASING STORAGE FOR CONTROL BLOCK** *cbname*

Explanation: An error occurred while attempting to release storage for the control block specified. The return code from the STORAGE RELEASE is provided.

returncode is the return code from the storage routine. *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for a description of the STORAGE RELEASE return codes.

cbname is the name of the control block for which the storage release failed. The possible names are:

- SATQ - SNMP Subagent TrapQ block
- SANQ - SNMP Subagent NetmQ block
- CDMQ - VARY/DISPLAY Command Request block
- CDMH - VARY/DISPLAY Command Header block

System Action: TCPIP continues.

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

System Programmer Response: This is an internal error. Contact the IBM Software Support Center.

EZZ3255I *taskname* **HAS NOT BEEN STARTED DUE TO ERROR** *returncode* **ATTEMPTING TO** *function*

Explanation: During TCPIP initialization, a failure occurred while attempting to start the component specified.

taskname is the name of the task that was to be started. The possible names are:

- EZACDMSM - Command Manager
- EZACFMMN - Configuration
- EZASASUB - SNMP Subagent

returncode is the return code from the system call specified.

function is the function that was attempted.

System Action: TCPIP continues or ends based on the task that failed to initialize. If the task is EZACFMMN or EZACDMSM, TCPIP was not initialized. If the task is EZASASUB, the SNMP Subagent will not be available but TCPIP is initialized.

User or Operator Response: Notify the system programmer.

System Programmer Response: Turn on ITRACE for the component specified and restart TCPIP. Gather the documentation and contact the IBM Software Support Center. For information about ITRACE, see the *z/OS Communications Server: IP Configuration Reference*.

EZZ3300I osnmp is unable to open message catalog ″**snmpclim.cat**″ **:** *error*

Explanation: osnmp was unable to open the osnmp message catalog, snmpclim.cat, in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be ″NLSPATH=/usr/lib/nls/msg/%L/%N″.

System Action: osnmp will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *z/OS C/C++ Run-Time Library Reference* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in *z/OS UNIX System Services Programming Tools* If the default messages are acceptable, no action is necessary.

Source Data Set: snmp

Procedure Name: main

EZZ3301I Error return from *api*

Explanation: The call to the specified routine failed. This is an internal error.

System Action: osnmp ends.

User or Operator Response: Reissue the command specifying **-d 4**. Collect the trace output and report the problem to the system programmer.

System Programmer Response: Check that TCPIP is running. Restart TCPIP if necessary. Make sure your system

is configured correctly. See *z/OS Communications Server: IP Configuration Reference* for configuration information. If your system is configured correctly then contact the IBM Software Support center for assistance.

Source Data Set: snmp

Procedure Name: main

EZZ3303I Error finding network address for *host*

Explanation: An IP address could not be obtained for the host specified by the **-h** parameter.

System Action: osnmp ends.

User or Operator Response: Verify the hostname is correct and reissue the command. If the error continues, reissue the command with the target IP address instead of the name. Notify your system programmer if the problem persists.

System Programmer Response: Determine if the name server and/or HOSTS.SITEINFO are correct. Refer to the *z/OS Communications Server: IP Configuration Guide* for more information about configuring the Domain Name System or the Site Table.

Source Data Set: snmp

Procedure Name: main

EZZ3304I Error finding local host name, errno = *errmsg* **Using loopback address 127.0.0.1**

Explanation: A call to gethostname() failed. The loopback IP address 127.0.0.1 will be used as the host address.

errmsg is a message that describes the error.

System Action: osnmp ends.

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

System Programmer Response: Check the configuration of your default TCP/IP transport provider. If your default transport provider is TCP/IP, check the TCPIP.DATA data set for a valid HOSTNAME keyword. If your default TCP/IP transport provider is AnyNet®, check the AnyNet ENVVAR data set for a valid HOSTNAME keyword. For more information about errnos, see *z/OS UNIX System Services Messages and Codes*.

Source Data Set: snmp

Procedure Name: main

EZZ3305I Error finding source local host address : *name*

Explanation: The IP address of the local host could not be obtained.

System Action: osnmp ends.

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

System Programmer Response: Check the configuration of your default TCP/IP transport provider. If your default transport provider is TCP/IP, check the TCPIP.PROFILE data set for a valid HOMELIST specification. If your default TCP/IP transport provider is AnyNet, check the AnyNet configuration initialization procedure for a valid ISTSKIFC command.

Source Data Set: snmp

Procedure Name: main

EZZ3306I Error converting *name* **to Entity**

Explanation: An error occurred when attempting to covert the destination name specified by the **-h** parameter to an internal destination Entity.

System Action: osnmp ends.

User or Operator Response: If the name specified by the **-h** parameter was not entered correctly, correct the error

and reissue the command. Otherwise, there may be a configuration problem. Notify your system programmer of the problem.

System Programmer Response: Check the configuration of your default TCP/IP transport provider. If your default transport provider is TCP/IP, check the TCPIP.DATA data set for a valid HOSTNAME keyword. If your default TCP/IP transport provider is AnyNet, check the AnyNet ENVVAR data set for a valid HOSTNAME keyword.

Source Data Set: snmp

Procedure Name: main

EZZ3307I Memory Allocation failed

Explanation: An allocation of C heap storage failed. There is insufficient memory to continue processing.

System Action: osnmp ends.

User or Operator Response: If the request contained a large list of variables and/or variable/value pairs, shorten the list and reissue the command.

System Programmer Response: None.

Source Data Set: snmp or wsnmpmgr

Procedure Name: various

EZZ3308I Set function requires variable name/value pair(s)

Explanation: When issuing an snmp set, a variable name and a variable value must be specified.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp **-?** for the correct syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3309I Only one variable allowed for *functionname* **function**

Explanation: More than one variable was specified with either the walk or the bulkwalk function. Only one is allowed.

System Action: The command ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp **-?** for the correct syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3310I Timeout after *number* **seconds**

Explanation: The response to the snmp request was not received before the timeout value was reached.

System Action: osnmp ends.

User or Operator Response: Reissue the command by setting a larger timeout value using the **-t** parameter. If the problem persists, contact your system programmer.

System Programmer Response: Determine if the snmp agent at the target is active. Start it if necessary. If the problem persists, issue the osnmp command with **-d** 4 debug and contact the IBM Software Support Center.

Source Data Set: snmp

Procedure Name: main

EZZ3311I Option *option* **was entered without a value.**

Explanation: The option specified was entered without providing a value.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp **-?** for the correct syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3312I *number* **is not a valid timeout value, ignored**

Explanation: The timeout value was not valid. Either it was not a number or it was zero.

System Action: osnmp continues. The parameter is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3313I Unsupported function: *name*

Explanation: A function was entered that is not known to the command.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp **-?** for the command syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3314I Function *name* **requires at least one variable**

Explanation: The function specified requires at least one variable, none were entered.

System Action: osnmp ends.

User or Operator Response: Correct the syntax and reissue the command. Issue osnmp **-?** for the command syntax.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3315I A *type* **request was received. The program is** *action*

Explanation: This message is displayed when either a termination signal or an interruption signal was received from LE/370 while osnmp was executing.

System Action: osnmp ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: termHand

EZZ3316I Command abended with *abendcode*, *reasoncode*.

Explanation: This message is displayed when a abend occurred while the osnmp command was in progress.

abendcode is the MVS Abend code. MVS abend codes are described in the *z/OS MVS System Codes*.

reasoncode is the reason code associated with this *abendcode*. They are listed in the description of the abend code in the *z/OS MVS System Codes*.

System Action: osnmp ends.

User or Operator Response: Correct the error indicated by *abendcode* and *reasoncode* and reissue the command.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: abndHand

EZZ3317I sigaction() failed for *signal* **:** *reason*

Explanation: osnmp encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, osnmp will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

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 may not be properly cleaned up.

SIGTERM
handler controls cleanup of resources during termination. If sigaction fails for SIGTERM, certain resources may not be properly cleaned up when a SIGTERM is received.

SIGINT handler controls cleanup of resources during interactive attention. If sigaction fails for SIGINT, certain resources may not be properly cleaned up when a SIGINT is received.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3318I Unrecognized SMI type : *type*

Explanation: **osnmp** encountered a MIB object value type, *type*, that is not in the **osnmp** list of SMI types either when creating an SNMP packet or when decoding a packet that was received.

System Action: osnmp ends.

User or Operator Response: Repeat the SNMP request that generated the unrecognized type and collect traces. For the **osnmp** command, use the **-d** 4 level of traces to ensure that packets are traced. If the error is encountered when the **osnmp** command receives a packet from an SNMP agent, collect traces from the originating SNMP agent that contains the generated packets. Report the problem to the system programmer.

System Programmer Response: Collect the traces and contact the IBM Software Support Center.

Source Data Set: snmp

Procedure Name: SnmpPrintValue

EZZ3319I *variable* **is not a valid variable.instance**

Explanation: The syntax of a variable name or ObjectID is not correct or a variable name, not an instance of the variable was entered. The correct syntax is one of the following:

- `varName.n`, where *varName* is alphanumeric, starting with an alphabetic (a-z) character (such as, sysDescr.0).
- `n.n....n`, where *n* is one or more digits (such as, 1.3.6.1.2.1.1.1.0).

System Action: osnmp ends.

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

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3320I **variable** *variable* **is not found in any local MIB**

Explanation: The variable specified was not found in either the MIB compiled into the osnmp command or in the MIBS.DATA file.

System Action: The command ends.

User or Operator Response: If the variable specified is incorrect, correct it and reissue the command. If the variable is correct, reissue the command using the object identifier instead of the textual name.

System Programmer Response: Consider adding additional objects to the MIBS.DATA file. See *z/OS Communications Server: IP Configuration Guide* for more information.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3321I **Error in** *MIBS.DATA* **file,** *lineno* **:** *error*

Explanation: While the command processor was reading the MIBS.DATA file, it encountered an error in the file on the line specified. The possible errors are :

line too long
Each line in the file must be no longer than 2048. The line specified is longer.

missing field(s)
One or more fields are missing from the file. The format of the file is:

```
character_object_name  object_identifier  object_type
```

unrecognized type
The value in the object_type field is not one of the expected types. Refer to the section about configuring the OSNMP command in the *z/OS Communications Server: IP Configuration Reference* for a list of object_types.

System Action: The line is ignored and osnmp continues.

User or Operator Response: Correct the error indicated and reissue the command.

System Programmer Response: None.

Source Data Set: snmp_mtable

Procedure Name: main

EZZ3322I *value* **is not numeric**

Explanation: The value must be numeric.

System Action: The command ends.

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

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3323I *value* **value is out of range**

Explanation: The value entered is outside the range defined by the command syntax or the SMI type.

System Action: The command ends.

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

System Programmer Response: None.

Source Data Set: wsnmpmgr, snmp

Procedure Name: snmp_fill_varBind, init_args

EZZ3324I *value* **is not a valid IP address**

Explanation: The IP address specified is not syntactically correct.

System Action: The command ends.

User or Operator Response: Correct the value and reissue the command.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3325I *value* **is not a valid OID**

Explanation: The OID specified is not syntactically correct.

System Action: The command ends.

User or Operator Response: Correct the value and reissue the command.

System Programmer Response: None.

Source Data Set: wsnmpmgr

Procedure Name: snmp_fill_varBind

EZZ3326I Variables cannot be specified with the TRAP function

Explanation: A variable was specified after the TRAP function. Variables are not applicable when specifying TRAP.

System Action: osnmp ends.

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

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3327I No valid PDUs returned on this *function* **request**

Explanation: A request was made to walk a MIB tree via a walk or a bulkwalk request. The next variable in the tree did not have the same prefix as the variable specified on the command. The PDU was thrown away.

System Action: osnmp ends.

User or Operator Response: Reissue the command specifying a valid MIB tree prefix.

System Programmer Response: None

Source Data Set: snmp

Procedure Name: main

EZZ3328I -p allowed only with TRAP function

Explanation: The **-p** option is valid only when specified with the TRAP function. The function specified was not TRAP.

System Action: osnmp ends.

User or Operator Response: Reissue the command with out the **-p** option.

System Programmer Response: None

Source Data Set: snmp

Procedure Name: main

EZZ3329I *portnumber* **is not a valid port number**

Explanation: Port numbers must be between 1 and 65535. The port number specified was not in the range.

System Action: osnmp ends.

User or Operator Response: Correct the port number and reissue the command.

System Programmer Response: None

Source Data Set: snmp

Procedure Name: main

EZZ3330I SET request failed. No values changed.

Explanation: A SET request failed. The object on which the failure occurred and the reason for the failure are indicated in the object list displayed prior to this message. If multiple objects were to be changed on the same SET request, no values will be changed if any of the objects are in error. The object list will, however, display the values passed in on the SET request, instead of the resulting values.

System Action: osnmp ends.

User or Operator Response: Reissue a valid SET command.

System Programmer Response: None

Source Data Set: snmp

Procedure Name: main

EZZ3331I The osnmp command is not available because the TCP/IP Base feature of OS/390 is not enabled.

Explanation: The osnmp command issued a query for product=OS/390, feature=TCP/IP_BASE but IFAPRDxx parmlib member does not indicate that the feature TCP/IP_BASE is enabled. This osnmp command will only run if TCP/IP_BASE is enabled.

System Action: The osnmp command is terminated.

User or Operator Response: Contact the system programmer.

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

Source Data Set: snmp

Procedure Name: main

EZZ3332I Only -d option is valid with findname.

Explanation: The user entered an invalid option with function findname. Only the -d option is allowed.

System Action: The osnmp command is terminated.

User or Operator Response: Reissue osnmp findname with only the **-d** (debug) as an allowable option.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: init_args

EZZ3333I osnmp is a UNIX System Services command and must be issued from an OS/390 shell.

Explanation: The osnmp command was not issued from an OS/390 shell.

System Action: The osnmp command terminates.

User or Operator Response: Reissue the command from an OS/390 shell. For information about the osnmp command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: snmp

Procedure Name: main

EZZ3334I Unrecognized *valuetype* **in file** *filename* **(Line** *linenumber***)**

Explanation: An unrecognized value was encountered while processing the specified configuration file at the indicated line.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3335I *Valuetype* **too long in file** *filename* **(Line** *linenumber***)**

Explanation: An error was encountered processing the specified configuration file at the indicated line. The value for the indicated field is too long.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3336I *Valuetype* **too short in file** *filename* **(Line** *linenumber***)**

Explanation: An error was encountered processing the specified configuration file at the indicated line. The value for the indicated field is too short.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp

command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3337I Entry in file *filename* **(Line** *linenumber***) should have at least** *number* **fields.**

Explanation: An entry in the configuration file should have at least as many fields as indicated. Different administrative models require different number of fields.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3338I *keytype* **in** *filename* **(Line** *linenumber***) has an incorrect length.**

Explanation: The indicated key is not the correct length.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3339I Error in *filename* **(Line** *linenumber***) - Line too long.**

Explanation: While the command processor was reading the command configuration file, it encountered a line that was too long.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3340I Error processing *filename* **(Line** *linenumber***).**

Explanation: The osnmp command encountered an error while processing the specified line.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3341I Line number *linenumber* **in the** *filename* **file has a network address or host name which is not valid.**

Explanation: The network address at the specified line number is not in dotted decimal notation(xxx.xxx.xxx.xxx), or the host name specified is not valid.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3342I Line number *linenumber* **in the** *filename* **file has a port number which is not valid.**

Explanation: The port number at the specified line number is not valid.

System Action: The line is ignored, and the osnmp command continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ws_confi

Procedure Name: win_db_read_config_file

EZZ3343I Report received : *exception counter MIB object*

Explanation: A report PDU was received, indicating than an exception condition occurred. The name of the MIB object incremented for this exception is displayed. Reports are typically issued when the configuration at the osnmp command is not consistent with the configuration at the target agent. For example, the user name in the request may not be configured at the agent or the authentication and/or privacy keys may not be valid.

System Action: The osnmp command displays error messages and ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. For information about configuring the osnmp command, refer to the *z/OS Communications Server: IP Configuration Guide*.

Source Data Set: ws_util

Procedure Name: sd_win_display_report

Chapter 4. EZZ4*xxxx* messages

EZZ4200I TCP/IP INITIALIZATION COMPLETE FOR *jobname*

Explanation: Initialization completed successfully for TCP/IP.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: The TCP/IP initialized.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBITINI

EZZ4201I TCP/IP TERMINATION COMPLETE FOR *jobname*

Explanation: This TCP/IP terminated.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: The TCP/IP procedure terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBITINI

EZZ4202I OPENEDITION-TCP/IP CONNECTION ESTABLISHED FOR *jobname*

Explanation: This TCP/IP established a connection with z/OS UNIX System Services (OpenEdition®).

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: The TCP/IP procedure established a connection with z/OS UNIX System Services.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBTIPFS

EZZ4203I OPENEDITION-TCP/IP CONNECTION ERROR FOR *jobname-function name,function type,return value,return code,reason code*

Explanation: This TCP/IP failed to established a connection with z/OS UNIX System Services (OpenEdition).

jobname is the jobname associated with the procedure used to start TCP/IP.

function name is the name of the function that failed while trying to establish the connection with z/OS UNIX System Services. See the *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for more information about the function that failed and the *return value*.

function type is the particular type for this function.

return value is the return value from the function attempted. See the *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for more information about the function that failed and the *return value*.

return code is the return code from the function attempted. See the *z/OS UNIX System Services Messages and Codes* for a list of the return codes.

reason code is the reason code from the function attempted. See the *z/OS UNIX System Services Messages and Codes* for a list of the reason codes.

System Action: The TCP/IP failed to initialize.

User or Operator Response: Contact the system programmer.

© Copyright IBM Corp. 1994, 2002

System Programmer Response: There are a few customization errors that could cause TCP/IP to not to be able to connect with z/OS UNIX System Services (OpenEdition). The TCP/IP jobname was not specified correctly and does not match one in the BPXPRMxx member that z/OS UNIX System Services is using. z/OS UNIX System Services was not configured with a CINET definition. The userid associated with the TCP/IP started procedure does not have the proper z/OS UNIX System Services authorization. Collect any supporting documentation and dumps, if available, and contact the IBM Software Support.

Source Data Set: EZBTIPFS

EZZ4204I TCPIP INITIALIZATION FOR *jobname* **FAILED.**

Explanation: Initialization failed for TCP/IP.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: This TCP/IP failed to initialize.

User or Operator Response: This message should be preceded by more specific error messages. Correct the errors indicated by the preceding messages.

System Programmer Response: This message should be preceded by more specific error messages. Correct the errors indicated by the preceding messages.

Source Data Set: EZBITINI

EZZ4205I TCPIP INITIALIZATION UNABLE TO START *jobname* **REASON:** *reason***.**

Explanation: This TCP/IP could not be started because one was already started with the same name, the maximum number of 8 TCP/IPs were already started, or a TCP/IP is already active in an INET configuration.

jobname is the jobname associated with the procedure used to start TCP/IP.

reason will be one of the following:

TCPIP ALREADY EXISTS
A TCP/IP with this jobname was already started on this system.

MAXIMUM OF 8 TCPIPS ALREADY STARTED
The maximum number of TCP/IPs have already been started on this system.

TCPIP ALREADY ACTIVE IN INET CONFIGURATION
A TCP/IP is already active. Only one TCP/IP can be active in an INET configuration.

System Action: This TCP/IP failed to initialize.

User or Operator Response: Determine if the correct TCP/IP jobname was used to start TCP/IP.

System Programmer Response: Take the action based on the specific failure explanation. The system programmer should either verify unique TCP/IP jobnames, verify that eight or less TCP/IPs have been started, or verify that common inet is properly coded in BPXPRMxx if a multiple stack environment is desirable.

Source Data Set: EZBITINI

EZZ4206I TCPIP INITIALIZATION FOR *jobname* **UNABLE TO OBTAIN CSM STORAGE. STORAGE TYPE:** *type***{DATASPACE|ECSA} RETURN CODE:** *return code* **REASON CODE:** *reason code*

Explanation: TCP/IP initialization failed trying to obtain CSM storage.

jobname is the jobname associated with the procedure used to start TCP/IP.

type is the storage type and will be one of the following:

DATASPACE
TCP/IP failed to obtain CSM dataspace storage.

ECSA TCP/IP failed to obtain CSM ECSA storage.

return code is the return code from the IVTCSM REQUEST(CREATE_POOL) macro invocation

reason code is the reason code from the IVTCSM REQUEST(CREATE_POOL) macro invocation

System Action: This TCP/IP will fail to initialize.

User or Operator Response: Determine from the return and reason codes from the CSM invocation as to why TCP/IP was unable to obtain the storage. See *z/OS Communications Server: CSM Guide* for return code documentation.

System Programmer Response: Determine why the storage could not be obtained. Make sure that VTAM was started and is at the appropriate level.

Source Data Set: EZBITINI

EZZ4207I TCPIP INITIALIZATION FOR *jobname* **ENCOUNTERED AN ENVIRONMENT ERROR. {NOT RUNNING AS A STARTED PROCEDURE.|NOT RUNNING IN KEY 6}**

Explanation: TCP/IP initialization failed because it detected it was not running in the correct environment.

jobname is the jobname associated with the procedure used to start TCP/IP.

error is one of the following:

NOT RUNNING AS A STARTED PROCEDURE.
TCP/IP detected it was not running as a started procedure.

NOT RUNNING IN KEY 6 TCP/IP detected it was not running in the correct key.

System Action: This TCP/IP will fail to initialize.

User or Operator Response: Response depends on the value of *error*:

NOT RUNNING AS A STARTED PROCEDURE.
Only try to run TCP/IP as a started procedure.

NOT RUNNING IN KEY 6 Make sure that all libraries in the linklist concatenation (including any STEPLIBs or JOBLIBs) are APF-authorized.

System Programmer Response: Response depends on the value of *error*:

NOT RUNNING AS A STARTED PROCEDURE.
Only try to run TCP/IP as a started procedure.

NOT RUNNING IN KEY 6 Make sure that all libraries in the linklist concatenation (including any STEPLIBs or JOBLIBs) are APF-authorized.

Source Data Set: EZBITINI

EZZ4208I BPX1SDD FAILED WITH RETURN CODE: *return code*, **REASON CODE:** *reason code*. **THE SNMP SUBAGENT IS NOT AVAILABLE.**

Explanation: A call to z/OS UNIX System Services routine BPX1SDD failed with the return and reason code provided. The SNMP subagent cannot be initialized and will not be available.

return code is the return code from BPX1SDD.

reason code is the reason code from BPX1SDD.

System Action: This TCP/IP continues.

User or Operator Response: Try to determine why BPX1SDD failed from the return code and the reason code provided. This information is found in the *z/OS UNIX System Services Programming: Assembler Callable Services Reference* publication.

System Programmer Response: Contact your IBM support center for assistance.

Source Data Set: EZBITINI

EZZ4209I CTRACE OPTION *ctrace option* **IS NOT VALID**

Explanation: In response to the TRACE CT® command one of the options value is not supported by TCP/IP.

ctrace option is the unrecognized CTRACE option.

System Action: The CTRACE request is terminated with message ITT004I.

User or Operator Response: Reissue the TRACE CT command and specify a valid option value. Supported trace options can be found in the *z/OS Communications Server: IP Diagnosis*.

Source Data Set: EZBCTSSM

EZZ4210I CTRACE DEFINE FAILED FOR *parmlib member name*, **RETURN CODE:** *return code* **REASON CODE:** *reason code*

Explanation: The request to define a component trace for TCP/IP or the Resolver failed. The return codes and reason codes are defined in: *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*.

parmlib member name is the CTRACE Parmlib member name.

return code is the return code from the CTRACE DEFINE macro invocation.

reason code is the reason code from the CTRACE DEFINE macro invocation.

System Action: For TCP/IP CTRACE component SYSTCPIP, if the parmlib member was not CTIEZB00, then a CTRACE DEFINE is attempted using CTIEZB00. If the parmlib member was CTIEZB00, then the CTRACE DEFINE is attempted with system defaults.

For the Resolver CTRACE component SYSTCPRE, the CTRACE DEFINE is attempted with system defaults.

User or Operator Response: Contact the system programmer to correct the parmlib member.

System Programmer Response: Correct the parmlib member and restart TCP/IP or the Resolver or issue the `TRACE CT,ON,COMP=SYSTCPIP,SUB=(`*`procedure_name`*`),PARM=`*`parmlib_member`* command to update the TCP/IP component trace functions or issue the `TRACE CT,ON,COMP=SYSTCPRE,SUB=(`*`procedure_name`*`),PARM=`*`parmlib_member`* command to update the Resolver component trace functions .

Source Data Set: EZBCTINI, EZBRECTI

EZZ4211I BLDL FOR MODULE EZBITCOM FAILED, RETURN CODE: *return code*

Explanation: A BLDL macro call to locate module EZBITCOM failed with the return code provided. TCP/IP cannot complete its initialization.

return code is the return code from the BLDL macro.

System Action: TCP/IP terminates.

User or Operator Response: Contact your system programmer.

System Programmer Response: Try to determine why the BLDL failed from the return code provided. This information is found in the *z/OS DFSMS Macro Instructions for Data Sets* publication. Verify that load module EZBITCOM exists in data set *hlq*.SEZALOAD, where *hlq* is your installation defined high level qualifier. You should also verify that *hlq*.SEZALOAD is either in the default MVS link list or that it is explicitly specified as a STEPLIB DD card on the started procedure JCL used to start this TCP/IP instance. If no problems are found, collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: EZBITINI

EZZ4212I LOAD FOR MODULE EZBITCOM FAILED, ABEND CODE: *abend code* **ABEND REASON CODE:** *abend reason code*

Explanation: A LOAD macro call for module EZBITCOM failed with the abend code and reason code provided. TCP/IP cannot complete its initialization.

abend code is the Abend Code from the LOAD macro.

abend reason code is the abend reason code from the LOAD macro invocation.

System Action: TCP/IP terminates.

User or Operator Response: Contact your system programmer.

System Programmer Response: Try to determine why the LOAD failed from the abend code and abend reason code provided. This information is found in the *z/OS MVS System Codes* publication. Verify that load module EZBITCOM exists in data set *hlq*.SEZALOAD, where *hlq* is your installation defined high level qualifier. You should also verify that *hlq*.SEZALOAD is either in the default MVS link list or that it is explicitly specified as a STEPLIB DD card on the started procedure JCL used to start this TCP/IP instance. If no problems are found, collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: EZBITINI

EZZ4213I COULD NOT OBTAIN CSA STORAGE, RETURN CODE: *return code*

Explanation: A STORAGE (OBTAIN) system call for CSA storage failed with the return code provided. TCP/IP cannot complete its initialization.

return code is the return code from the STORAGE (OBTAIN) macro.

System Action: TCP/IP terminates.

User or Operator Response: Contact your system programmer.

System Programmer Response: Try to determine why the STORAGE (OBTAIN) call failed from the return code provided. This information is found in the *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO* publication. If the return code indicates that insufficient storage or system resources were available, correct the problem and restart TCP/IP. Otherwise, collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: EZBITINI

EZZ4214I STORAGE RELEASE FOR CSA FAILED, RETURN CODE: *return code*

Explanation: An unexpected internal error occurred while trying to free CSA storage. The return code from the STORAGE (RELEASE) system service is provided.

return code is the return code from the STORAGE (RELEASE) macro.

System Action: TCP/IP Continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: EZBITINI

EZZ4215I TCP/IP ABEND - DUMPING

Explanation: TCP/IP recovery was entered, and a dump will be taken.

System Action: TCP/IP Continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: EZBITDRC

EZZ4216I TCP/IP ABEND, DUMP SUPPRESSED DUE TO A PREVIOUS ERROR

Explanation: TCP/IP recovery was entered, but no dump will be taken due to a previous abend by this work unit.

System Action: TCP/IP Continues.

User or Operator Response: Contact your system programmer.

System Programmer Response: Collect any available supporting documentation and dumps, and contact the IBM Software Support Center.

Source Data Set: EZBITDRC

EZZ4217I SYNTAX ERROR IN PARAMETERS SPECIFIED ON PARMS KEYWORD FOR *jobname*

Explanation: A syntax error was detected on the PARMS= statement in the started task procedure for TCP/IP. Message EZZ4218I follows this message with an indication of where the syntax error was detected. All parameters after the syntax error are ignored. Parameters prior to the syntax error were processed successfully.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: TCP/IP continues its initialization process.

User or Operator Response: Correct the parameters in error and restart TCP/IP if necessary.

System Programmer Response: Correct the parameters in error and restart TCP/IP if necessary.

Source Data Set: EZBITINI

EZZ4218I PARAMETERS BEGINNING WITH *param* **WILL BE IGNORED**

Explanation: A syntax error was detected in the parameter specified in the message text. All parameters after the parameter in error are ignored. This is the next message after the message EZZ4217I. Please see EZZ4217I for the further information.

param is the parameter in error in the PARMS= statement of the started task procedure for TCP/IP.

System Action: TCP/IP continues its initialization process.

User or Operator Response: Correct the parameters in error and restart TCP/IP if necessary.

System Programmer Response: Correct the parameters in error and restart TCP/IP if necessary.

Source Data Set: EZBITINI

EZZ4219I ASSEMBLER SERVICE *service_name* **COMPLETED WITH RETURN CODE** *return_code* **REASON CODE** *reason_code*

Explanation: The specified system service completed with a non-zero return code. The hexadecimal return and reason code are shown.

service_name is the name of the assembler service.

return_code is the return code of the called service.

reason_code is the reason code of the called service.

System Action: TCP/IP continues if possible.

User or Operator Response: Check the system service return and reason code in the *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*. When the problem is resolved, restart TCP/IP if it terminated.

System Programmer Response: Correct the error as indicated by the *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*. For the service name DSPSERVE CREATE and reason code 6C000611, a SCOPE=COMMON request failed. The IEASYSxx parameter MAXCAD limits the number of SCOPE=COMMON dataspaces available.

Source Data Set: EZBCTINI

EZZ4221I ARM FUNCTION *function name* **FAILED FOR** *jobname*, **RC=***return code*, **RSN=***reason code*

Explanation: The TCP/IP stack, whose name appears in the message, issued the named Automatic Restart Manager function but the function failed.

function name is the name of the ARM function that failed.

jobname is the jobname associated with the procedure used to start TCP/IP.

return code is the return code from the function attempted.

reason code is the reason code from the function attempted.

System Action: TCP/IP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: The return code and reason code values in the message belong to the MVS Automatic Restart Manager and are documented under the IXCARM macro in the *z/OS MVS Programming: Sysplex Services Reference* manual. Correct the problem indicated by the return and reason codes.

Source Data Set: EZBTIINI

EZZ4300I CLAW DEVICE *device_name*: **INCORRECT ADAPTER MICROCODE VERSION** *version*

Explanation: An incorrect version of the microcode was loaded onto the CLAW adapter in the workstation.

device_name is the name of the device.

version is the version level of the adapter microcode.

System Action: TCPIP does not start the CLAW device.

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

System Programmer Response: Obtain the correct level of the CLAW microcode and restart the CLAW adapter. Restart the CLAW device.

EZZ4301I CLAW DEVICE *device_name*: **RECEIVED HOST NAME** *received_host_name* **ADAPTER NAME** *received_adapter_name*, **EXPECTED HOST NAME** *expected_host_name* **ADAPTER NAME** *expected_adapter_name*

Explanation: The CLAW device parameters in the TCPIP profile do not match the CLAW adapter configuration parameters.

device_name is the name of the device.

received_host_name is the host name received from the CLAW device.

received_adapter_name is the adapter name received from the CLAW device.

expected_host_name is the host name expected from the CLAW device.

expected_adapter_name is the adapter name expected from the CLAW device.

System Action: TCPIP does not start the CLAW device.

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

System Programmer Response: Correct the host and adapter name parameters in either the CLAW adapter configuration or the CLAW DEVICE statement in the TCPIP profile. Use the VARY TCPIP command to restart the CLAW device.

EZZ4302I CLAW DEVICE *device_name*: **ADAPTER RECEIVE SIZE OF** *adapter_receive_size* **IS SMALLER THAN HOST TRANSMIT SIZE OF** *host_transmit_size*

Explanation: The CLAW device parameters in the TCPIP profile do not match the CLAW adapter configuration parameters. The write buffer size on the CLAW DEVICE statement is larger than the CLAW adapter receive buffer size.

device_name is the name of the device.

adapter_receive_size is the size of data that the CLAW device can receive.

host_transmit_size is the size of data that TCP/IP can send to the CLAW device.

System Action: TCPIP does not start the CLAW device.

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

System Programmer Response: Correct the buffer size in either the CLAW adapter configuration or the CLAW

DEVICE statement in the TCPIP profile. Use the VARY TCPIP command to restart the CLAW device.

EZZ4303I CLAW DEVICE *device_name***: ADAPTER TRANSMIT SIZE OF** *adapter_transmit_size* **IS GREATER THAN HOST RECEIVE SIZE OF** *host_receive_size*

Explanation: The CLAW device parameters in the TCPIP profile do not match the CLAW adapter configuration parameters. The read buffer size on the CLAW DEVICE statement is smaller than the CLAW adapter transmit buffer size.

device_name is the name of the device.

adapter_transmit_size is the size of data that the CLAW device can send.

host_receive_size is the size of data that TCP/IP can receive from the CLAW device.

System Action: TCPIP does not start the CLAW device.

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

System Programmer Response: Correct the buffer size in either the CLAW adapter configuration or the CLAW DEVICE statement in the TCPIP profile. Use the VARY TCPIP command to restart the CLAW device.

EZZ4304I LCS DEVICE *device_name***: ERROR** *lanstats_error_code* **IN GETTING HOME HARDWARE ADDRESS FOR LINK** *link_name***. CHECK LINK_NUMBER ON LINK STATEMENT.**

Explanation: The LCS device reported that no LAN adapter is physically installed in the slot identified by link_number on the TCPIP LINK statement.

device_name is the name of the device.

lanstats_error_code is the error code as reported on the LCS Lanstats flow.

link_name is the name of the link.

System Action: TCPIP does not initialize the LCS link.

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

System Programmer Response: Verify that the link_number, as specified on the LINK statement, matches the slot position within the LCS device. If necessary, correct the link_number parameter on the LINK statement, and restart the LCS device. Refer to the LCS device documentation for more information about the *lanstats error code*.

EZZ4305I UNABLE TO RECOVER DEVICE *device_name*

Explanation: Recovery attempts for the device were unsuccessful. The original error (which initiated the recovery action) is described in message EZZ4310I. Either message EZZ4306I or EZZ4307I will also accompany this message, describing why recovery attempts were abandoned.

device_name is the name of the device.

System Action: The device is left in the Not Ready state.

User or Operator Response: Perform the action described for EZZ4310I.

System Programmer Response: Perform the action described for EZZ4310I.

EZZ4306I REASON: REACHED UNSUCCESSFUL RETRY THRESHOLD

Explanation: This message accompanies EZZ4305I and EZZ4310I or EZZ4346I and EZZ4338I. Recovery attempts were unsuccessful at returning the device to the Ready state.

System Action: The device is left in the Not Ready state.

User or Operator Response: Perform the action described for EZZ4310I or EZZ4338I.

System Programmer Response: Perform the action described for EZZ4310I or EZZ4338.

EZZ4307I REASON: ERROR ENCOUNTERED AFTER REACTIVATION

Explanation: This message accompanies EZZ4305I and EZZ4310I or EZZ4346I and EZZ4338I. Recovery attempts were successful at returning the device to the Ready state, but an error was detected on the device immediately following reactivation.

System Action: The device is left in the Not Ready state.

User or Operator Response: Perform the action described for EZZ4310I or EZZ4338I.

System Programmer Response: Perform the action described for EZZ4310I or EZZ4338I.

EZZ4308I ERROR: CODE=*error_code* **DURING** *link_control_function* **DEVICE** *device_name***. DIAGNOSTIC CODE:** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during activation of the device.

error_code is the status code for the link layer.

link_control_function is the function being performed on the device.

device_name is the name of the device.

internal_diagnostic_code is the internal diagnostic code for use by IBM.

System Action: TCP/IP deactivates the device.

User or Operator Response: Consult the Data Link Control (DLC) Status Codes chapter in *z/OS Communications Server: IP and SNA Codes* for a description of the status code for the link layer. If applicable, correct the hardware problem and restart the device. If the last 4 digits if the error code are 3332, then the most likely reason is that one of the devices is offline. If the last 4 digits if the error code are 3016, then the most likely reason is that the TRLE definition for the device is not active.

System Programmer Response: Perform the action described in *z/OS Communications Server: IP and SNA Codes* for the indicated status code.

EZZ4309I ATTEMPTING TO RECOVER DEVICE *device_name*

Explanation: An error was detected on the device, as reported by message EZZ4310I, and the link layer is attempting to recover the device. If the recovery attempt is successful, either message EZZ4313I or EZZ4314I will accompany this message. If the recovery attempt is unsuccessful, message EZZ4305I will be issued.

device_name is the name of the device.

System Action: TCP/IP attempts to recover the device.

User or Operator Response: If accompanied by message EZZ4305I, device recovery was unsuccessful. Perform the action described for EZZ4310I. If accompanied by EZZ4313I or EZZ4314I, recovery of the device was successful, and no further action is necessary. If neither EZZ4305I nor EZZ4313I/EZZ4314I is issued, TCP/IP is awaiting a ″ready″ indication from the device. (This is typically seen on CTC devices, where the remote TCP/IP image is either down or has not yet issued a START DEV for the CTC connection.)

EZZ4310I ERROR: CODE=*error_code* **REPORTED ON DEVICE** *device_name***. DIAGNOSTIC CODE:** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during operation of the device. The error was reported asynchronous to execution of any Link Control function.

error_code is the status code for the link layer.

device_name is the name of the device.

internal_diagnostic_code is the Internal Diagnostic for use by IBM.

System Action: TCP/IP marks the device inactive.

User or Operator Response: Consult the Data Link Control (DLC) Status Codes section in *z/OS Communications Server: IP and SNA Codes* for a description of the status code for the link layer. If applicable, correct the hardware problem and restart the device.

System Programmer Response: Perform the action described in *z/OS Communications Server: IP and SNA Codes* for the indicated status code.

EZZ4311I LINK *link_name* **HAS FAILED ON DEVICE** *device_name*

Explanation: TCPIP received a signal from the LAN adapter indicating that this link is not available. This can occur if a cable becomes unplugged.

link_name is the name of the link.

device_name is the name of the device.

System Action: TCPIP attempts to reactivate the link.

User or Operator Response: Check the LAN adapter for this link and ensure that all cables are properly connected. If necessary, stop and restart the device.

System Programmer Response: None

EZZ4312I LCS DEVICE *device_name***: ERROR** *start_lan_return_code* **IN STARTING LAN ADAPTER FOR LINK** *link_name*

Explanation: The LAN adapter was not successfully started. Possible reasons are:

- There is an error on the LINK statement in the TCPIP profile.
- The LAN adapter is not connected to a live LAN.

device_name is the name of the device.

start_lan_return_code is the return code on LCS Start LAN.

link_name is the name of the link.

System Action: TCPIP does not initialize the LCS link.

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

System Programmer Response:

- Verify that the link_number on the LINK statement matches the slot position within the LCS device. If necessary, correct the link_number parameter on the LINK statement, and restart the LCS device.
- Verify that the LAN adapter is connected to a live LAN.

Refer to the documentation for your LCS device for more information about the *start lan return code*.

EZZ4313I INITIALIZATION COMPLETE FOR DEVICE *device_name*

Explanation: TCPIP successfully started the device.

device_name is the name of the device.

System Action: The device is ready for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

EZZ4314I INITIALIZATION COMPLETE FOR DEVICE *device_name***, LINK** *link_name*

Explanation: TCPIP successfully started the specified link for the specified device.

device_name is the name of the device.

link_name is the name of the link.

System Action: The link is ready for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

EZZ4315I DEACTIVATION COMPLETE FOR DEVICE *device_name*

Explanation: TCPIP deactivated the device.

device_name is the name of the device.

System Action: The device and its links are no longer available for use with TCPIP.

User or Operator Response: If the device is needed by TCPIP, use the VARY TCPIP command to restart the device.

System Programmer Response: None

EZZ4316I NO HOME IP ADDRESS DEFINED FOR CDLC DEVICE *device_name*

Explanation: TCPIP cannot setup the CDLC link because there is no local IP address defined for the CDLC link.

device_name is the name of the device.

System Action: TCPIP deactivates the CDLC device.

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

System Programmer Response: Correct the TCPIP profile to specify a home entry for the CDLC link. Use the VARY TCPIP command to update the home list and restart the CDLC device.

EZZ4317I INITIALIZATION COMPLETE FOR PVC *pvc_name* **FOR DEVICE** *device_name*

Explanation: TCPIP successfully activated the specified PVC for the specified device.

pvc_name is the name of the permanent virtual circuit.

device_name is the name of the device.

System Action: The PVC connection is ready for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

EZZ4318I DEACTIVATION COMPLETE FOR PVC *pvc_name* **FOR DEVICE** *device_name*

Explanation: TCPIP successfully deactivated the specified PVC for the specified device in response to a DELETE ATMPVC statement.

pvc_name is the name of the permanent virtual circuit.

device_name is the name of the device.

System Action: The PVC is no longer available for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

EZZ4319I ERROR: CODE=*error_code* **DURING** *link_control_function* **PVC** *pvc_name* **FOR DEVICE** *device_name***. DIAGNOSTIC CODE:** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during activation of the PVC.

error_code is the status code for the link layer.

link_control_function is the function being performed on the PVC.

pvc_name is the name of the PVC.

device_name is the name of the device.

internal_diagnostic_code is the Internal Diagnostic for use by IBM.

System Action: TCP/IP does not activate the PVC.

User or Operator Response: Consult the Data Link Control (DLC) Status Codes chapter in *z/OS Communications Server: IP and SNA Codes* for a description of the status code for the link layer. If applicable, correct the hardware problem and restart the ATM device to activate the PVC.

System Programmer Response: Perform the action described in *z/OS Communications Server: IP and SNA Codes* for the indicated status code.

EZZ4320I ERROR: CODE=*error_code* **DURING** *link_control_function* **SVC CONNECTION TO** *ip_address* **FOR DEVICE** *device_name***. DIAGNOSTIC CODE:** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during activation of the SVC for an ATM device.

error_code is the status code for the link layer.

link_control_function is the function being performed on the SVC.

ip_address is the destination IP address.

device_name is the name of the device.

internal_diagnostic_code is the Internal Diagnostic for use by IBM.

System Action: TCP/IP does not activate the SVC.

User or Operator Response: If the diagnostic code is a value other than 4, then consult the Data Link Control (DLC) Status Codes chapter in *z/OS Communications Server: IP and SNA Codes* for a description of the status code for the link layer. If applicable, correct the hardware problem and restart the ATM device. If the diagnostic code is 4, then the first byte of the error code is a Coding Standard, the third byte of the error code is a Cause Location, and the fourth byte of the error code is a hexadecimal Cause Code. Consult the ATM Network-Generated Cause and Diagnostic Codes chapter in *z/OS Communications Server: IP and SNA Codes* for a description of these codes. If applicable, correct the hardware problem and restart the ATM device.

System Programmer Response: Perform the action described in *z/OS Communications Server: IP and SNA Codes* for the indicated status code.

EZZ4321I ERROR: CODE=*error_code* **DURING** *link_control_function* **CONNECTION TO** *ip_address* **FOR DEVICE** *device_name***. DIAGNOSTIC CODE:** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during activation of a VC for an MPCPTP device.

error_code is the status code for the link layer.

link_control_function is the function being performed on the VC.

ip_address is the destination IP address.

device_name is the name of the device.

internal_diagnostic_code is the Internal Diagnostic for use by IBM.

System Action: TCP/IP does not activate the VC.

User or Operator Response: Consult the Data Link Control (DLC) Status Codes chapter in *z/OS Communications Server: IP and SNA Codes* for a description of the status code for the link layer. If applicable, correct the hardware problem and restart the MPCPTP device.

System Programmer Response: Perform the action described in *z/OS Communications Server: IP and SNA Codes* for the indicated status code.

EZZ4322I CONNECTION CLEARED FOR PVC *pvc_name***, DEVICE** *device_name***, CODE=** *cause_code***, ATTEMPTING TO RECOVER**

Explanation: TCPIP been notified that the specified PVC connection for the specified device is no longer active.

pvc_name is the name of the permanent virtual circuit.

device_name is the name of the device.

cause_code is the Clear Cause Code.

System Action: TCPIP will attempt to recover the PVC connection.

User or Operator Response: None.

System Programmer Response: None

EZZ4323I CONNECTION TO *ip_address* **CLEARED FOR DEVICE** *device_name*

Explanation: TCPIP been notified that the MPCPTP connection to the specified IP address for the specified device is no longer active.

ip_address is the destination IP address.

device_name is the name of the device.

System Action: The specified connection is no longer available for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

EZZ4324I CONNECTION TO *ip_address* **ACTIVE FOR DEVICE** *device_name*

Explanation: The MPCPTP connection to the specified IP address for the specified device is now active.

ip_address is the destination IP address.

device_name is the name of the device.

System Action: The specified connection is now available for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None

EZZ4325I PVC CONNECTION CLEARED - *pvc_name* **FOR DEVICE** *device_name* **HAS DUPLICATE IP ADDRESS** *ip_address*

Explanation: TCPIP cleared a PVC connection having a duplicate IP address.

pvc_name is the name of the permanent virtual circuit.

device_name is the name of the device.

ip_address is the destination IP address.

System Action: The PVC connection has been cleared.

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

System Programmer Response: Correct the duplicate IP address.

EZZ4326I ERROR *start_lan_error_code* **STARTING LAN ADAPTER FOR DEVICE** *device_name*

Explanation: The adapter reported an error during activation of the MPCIPA device.

start_lan_error_code is the error code reported by the adapter.

device_name is the name of the device.

System Action: TCP/IP does not start the MPCIPA device. If the return code is X'E080', then TCP/IP delays device activation until the LAN comes online. Otherwise, TCP/IP deactivates the MPCIPA device.

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

System Programmer Response: See the *zSeries: OSA-Express Customer's Guide and Reference* for information about the OSA Reject Codes and a description of the error. Ensure that the OSA configuration is correct. If necessary, correct the OSA configuration and restart the device.

EZZ4327I ERROR *setip_error_code* **REGISTERING IP ADDRESS** *ip_address* **FOR DEVICE** *device_name*

Explanation: The adapter reported an error attempting to register an IP address with the device.

setip_error_code is the error code reported by the adapter.

ip_address is the IP address.

device_name is the name of the device.

System Action: If this is the home IP address for this MPCIPA device, then TCP/IP does not start the MPCIPA device.

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

System Programmer Response: See the *zSeries: OSA-Express Customer's Guide and Reference* for information about the OSA Reject Codes and a description of the error. If the error indicates that the IP address is already in use, then change one of your IP addresses so that all IP addresses are unique and restart the device.

EZZ4328I ERROR *setrouting_error_code* **SETTING ROUTING FOR DEVICE** *device_name*

Explanation: The adapter reported an error attempting to register this TCP/IP instance as the primary or secondary router for this MPCIPA device. The most likely reason is that the routing attribute (PRIROUTER or SECROUTER) specified in the TCPIP profile has already been registered for this device by another TCP/IP instance.

setrouting_error_code is the error code reported by the adapter.

device_name is the name of the device.

System Action: TCP/IP starts the device, but with the NONROUTER attribute.

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

System Programmer Response: See the *zSeries: OSA-Express Customer's Guide and Reference* for information about the OSA Reject Codes and a description of the error. If the error indicates that another TCP/IP instance has already registered for this routing attribute, then check the TCPIP profiles and ensure that only one TCP/IP instance using this device is defined as PRIROUTER and only one TCPIP instance is defined as SECROUTER. Correct the TCPIP profiles and restart the device.

EZZ4329I LINK *takeover_link_name* **HAS TAKEN OVER ARP RESPONSIBILITY FOR INACTIVE LINK** *inactive_link_name*

Explanation: A link became inactive and TCP/IP detected another active link on the same physical network, which can take over ARP responsibility for the inactive link.

takeover_link_name is the name of the takeover link.

inactive_link_name is the name of the inactive link.

System Action: TCP/IP assigns ARP responsibility for the inactive link to the takeover link. TCP/IP sends a gratuitous ARP for the IP address of the inactive link and uses the takeover link to reply to ARP requests on behalf of the inactive link. If the inactive link becomes active again, then TCP/IP will reassign the ARP responsibility to that link.

User or Operator Response: None.

System Programmer Response: None.

EZZ4330E TCP/IP DEVICE START(S) WAITING FOR VTAM

Explanation: TCP/IP cannot process Start Device commands until VTAM initialization completes.

System Action: TCP/IP queues all Start Device commands and will process them when VTAM initialization completes.

User or Operator Response: Start VTAM.

EZZ4331I ERROR ESTABLISHING PACKED CLAW CONTROL LINK FOR DEVICE *dev_name* **– ATTEMPTING TO ACTIVATE IN UNPACKED MODE**

Explanation: The RESPONSE or CONFIRM signal from the device indicated the device does not support CLAW Packing.

dev_name is the name of the device.

System Action: TCP/IP invokes CLAW connection establishment in UNPACKED mode.

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

System Programmer Response: If the device does support CLAW Packing, review and correct the CLAW configuration statements within the device. If the device does not support CLAW Packing, remove the PACKED keyword from the z/OS CLAW Device Statement.

Procedure Name: main

EZZ4332I ERROR *errorcode* **DURING PACKED CLAW SIGNALLING FOR DEVICE dev_name – DEACTIVATING THE DEVICE**

Explanation: The RESPONSE or CONFIRM signal from the device indicated an unrecoverable error during packed CLAW signalling. The device is deactivated.

errorcode is the error code reported on the Packed CLAW control flow.

dev_name is the name of the device.

System Action: TCP/IP returns the device to the INACTIVE state.

User or Operator Response: Inform the system programmer about this message and the error code received.

System Programmer Response: For error code hex B4, verify that the device is configured with a CLAW MTU of 4092. For device error codes other than hex B4, contact the vendor that supplied the CLAW device.

Procedure Name: main

EZZ4333I ERROR INSTALLING REPLACEABLE STATIC ROUTE TO *dest* **USING** *gateway* **MASK** *mask* **- REASON** *reason*

Explanation: Because all dynamic routes to *dest* were removed, the IP layer attempted to add to the routing table a replaceable static route, which uses *gateway* and *mask*, to *dest* . The add failed for the reason given.

dest is the IP address of the route destination.

gateway is the IP address of the route gateway, 0.0.0.0 if directly connected.

mask is the subnet mask of the destination.

reason is the reason the add failed and is one of the following:

1 The gateway that the route uses is unreachable.

2 The interface that the route uses is not valid.

3 The route was defined incorrectly.

System Action: The failing route is not added to the TCP/IP routing table, and processing continues. Other defined replaceable static routes might be added to the TCP/IP routing table. The definition of the failed replaceable static route is retained and TCP/IP might attempt to add it again, if there are no other ways to reach the destination.

User or Operator Response: Contact the system programmer.

System Programmer Response: Response depends on reason code:

1 This might be a temporary condition that can resolve itself when the gateway becomes reachable. If network actions are needed to make the gateway reachable, perform them (for example bringing up routers or connections). Otherwise, check the ROUTE statement for the replaceable static route, and if the gateway field is incorrect, fix it.

2 Verify the interface name used on the ROUTE statement of the replaceable static route.

3 Correct invalid syntax on the ROUTE statement of the replaceable static route.

Source Data Set: IPV4RTE

Procedure Name: InstallReplRoutes

EZZ4334I ERROR SETTING VLAN USER PRIORITY FOR DEVICE *devicename*

Explanation: An OSA-Express error occurred while setting the VLAN user priority for the device.

System Action: The device remains active but VLAN user priorities are not used.

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

System Programmer Response: Contact OSA express service support

Source Data Set: ProcessIpaV1ControlPacket

Procedure Name: EZBIFIND

EZZ4335I SETTING VLAN USER PRIORITY NOT SUPPORTED FOR DEVICE *devicename*

Explanation: Cannot set the VLAN user priority for this device. The OSA-Express device does not have the correct microcode level.

System Action: The device remains active but VLAN user priorities are not used.

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

System Programmer Response: Contact OSA-Express support

Source Data Set: EZBIFSVL

Procedure Name: EZBIFIND

EZZ4336I ERROR DURING *link_control_function* **INTERFACE** *interface_name* **- CODE** *error_code* **DIAGNOSTIC CODE** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during activation of the interface.

link_control_function is the function being performed on the interface.

interface_name is the name of the interface.

error_code is the Data Link Control (DLC) status code for the link layer.

internal_diagnostic_code is an internal diagnostic code for use by IBM.

System Action: TCP/IP deactivates the interface.

User or Operator Response: If the last 4 digits of the error code are X'3016', the most likely reason for the error is that the TRLE definition for the interface is not active. In this case, activate the TRLE and restart the interface. Otherwise, inform the system programmer about the error.

System Programmer Response: See the *z/OS Communications Server: IP and SNA Codes* for information about Data Link Control (DLC) status codes for the link layer and perform the action described for the indicated status code. If applicable, correct the hardware problem and restart the interface.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4337I ATTEMPTING TO RECOVER INTERFACE *interface_name*

Explanation: An error was detected on the interface, as reported by message EZZ4338I, and the link layer is attempting to recover the interface. If the recovery attempt is successful, message EZZ4340I will accompany this message. If the recovery attempt is unsuccessful, message EZZ4346I will be issued.

interface_name is the name of the interface.

System Action: TCP/IP attempts to recover the interface.

User or Operator Response: If accompanied by message EZZ4346I, interface recovery was unsuccessful. Perform the action described for EZZ4338I. If accompanied by EZZ4340I, recovery of the interface was successful, and no further action is necessary.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4338I ERROR REPORTED ON INTERFACE *interface_name* **- CODE** *error_code* **DIAGNOSTIC CODE** *internal_diagnostic_code*

Explanation: The Link Layer detected an error during the operation of the interface. The error was reported asynchronous to execution of any Link Control function.

interface_name is the name of the interface.

error_code is the Data Link Control (DLC) status code for the link layer.

internal_diagnostic_code is an internal diagnostic code for use by IBM.

System Action: TCPIP marks the interface inactive.

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

System Programmer Response: See the *z/OS Communications Server: IP and SNA Codes* for information about Data Link Control (DLC) status codes for the link layer and perform the action described for the indicated status code. If applicable, correct the hardware problem and restart the interface.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4339I INTERFACE *interface_name* **FAILED - ADAPTER SIGNAL RECEIVED**

Explanation: TCPIP received a signal from the LAN adapter indicating that this interface is not available. One reason for this message is that a cable became unplugged.

interface_name is the name of the interface.

System Action: TCPIP attempts to restart the LAN adapter.

User or Operator Response: Check the LAN adapter for this link and ensure that all cables are properly connected. If necessary, stop and restart the interface.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4340I INITIALIZATION COMPLETE FOR INTERFACE *interface_name*

Explanation: TCPIP successfully started the interface.

interface_name is the name of the interface.

System Action: The interface is ready for use with TCPIP.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4341I DEACTIVATION COMPLETE FOR INTERFACE *interface_name*

Explanation: TCPIP deactivated the interface.

interface_name is the name of the interface.

System Action: The interface is no longer available for use with TCPIP.

User or Operator Response: If the interface is needed by TCPIP, use the VARY TCPIP command to restart the interface.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4342I ERROR *start_lan_error_code* **STARTING LAN ADAPTER FOR INTERFACE** *interface_name*

Explanation: The adapter reported an error during activation of the interface.

start_lan_error_code is the error code reported by the adapter.

interface_name is the name of the interface.

System Action: If the error code is X'E080', TCP/IP delays interface activation until the LAN comes online. Otherwise, TCP/IP deactivates the interface.

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

System Programmer Response: See the OSA Reject Codes section in the *zSeries: OSA-Express Customer's Guide and Reference* for a description of the error. Ensure that the OSA configuration is correct. If necessary, correct the OSA configuration and restart the interface.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4343I ERROR *setip_error_code* **REGISTERING IP ADDRESS** *ip_address* **FOR INTERFACE** *interface_name*

Explanation: The adapter reported an error while attempting to register an IP address with the interface.

setip_error_code is the error code reported by the adapter.

ip_address is the IP address.

interface_name is the name of the interface.

System Action: If this is the link local IP address for this interface, TCP/IP deactivates the interface. Otherwise, TCP/IP leaves the interface active.

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

System Programmer Response: See the OSA Reject Codes section in the *zSeries: OSA-Express Customer's Guide and Reference* for a description of the error code. If the error indicates that the IP address is already in use, ensure that your IP addresses are unique and restart the interface.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4344I ERROR *setrouting_error_code* **SETTING ROUTING FOR INTERFACE** *interface_name*

Explanation: The adapter reported an error while attempting to register this TCP/IP instance as the primary or secondary router for this interface. The most likely reason for the error is that the routing attribute (PRIROUTER or SECROUTER) specified in the TCPIP profile has already been registered for this interface by another TCP/IP instance.

setrouting_error_code is the error code reported by the adapter.

interface_name is the name of the interface.

System Action: TCP/IP starts the interface, but with the NONROUTER attribute.

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

System Programmer Response: See the OSA Reject Codes section in the *zSeries: OSA-Express Customer's Guide and Reference* for a description of the error code. If the error indicates that another TCP/IP instance has already registered for this routing attribute, check the TCPIP profiles and ensure that only one TCP/IP instance using this interface is defined as PRIROUTER and only one TCP/IP instance is defined as SECROUTER. Correct the TCPIP profiles and restart the interface.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4345I INTERFACE *takeover_interface_name* **HAS TAKEN OVER NED RESPONSIBILITY FOR INACTIVE INTERFACE** *inactive_interface_name*

Explanation: An interface became inactive and TCP/IP detected another active interface on the same physical network that can take over Neighbor Discovery (NED) responsibility for the inactive interface.

takeover_interface_name is the name of the takeover interface.

inactive_interface_name is the name of the inactive interface.

System Action: TCP/IP assigns NED responsibility for the inactive interface to the takeover interface. TCP/IP sends a gratuitous neighbor advertisement for the IP address of the inactive interface and uses the takeover interface to reply to neighbor solicitation requests on behalf of the inactive interface. If the inactive interface becomes active again, TCP/IP will reassign the NED responsibility to that interface.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4346I UNABLE TO RECOVER INTERFACE *interface_name*

Explanation: Recovery attempts for the interface were unsuccessful. The original error, which initiated the recovery action, is described in message EZZ4338I. Either message EZZ4306I or EZZ4307I will also accompany this message, describing why recovery attempts were abandoned.

interface_name is the name of the interface.

System Action: The interface remains inactive.

User or Operator Response: Perform the action described for EZZ4338I.

System Programmer Response: Perform the action described for EZZ4338I.

Source Data Set: TCPIP

Procedure Name: EZBIFIUT

EZZ4347I INTERFACE *interface_name* **DOES NOT SUPPORT IPV6**

Explanation: TCPIP cannot activate this interface because the adapter does not support IPv6.

interface_name is the name of the interface.

System Action: The interface remains inactive.

User or Operator Response: Contact system programmer.

System Programmer Response: Perform one of the following actions:

- Ensure that the adapter that corresponds to this INTERFACE definition supports IPv6 and restart the interface.
- Change the TCPIP profile to remove the INTERFACE definition.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ4348I ERROR INSTALLING REPLACEABLE STATIC ROUTE TO *dest/prefix_len* **USING** *gateway* **- REASON** *reason*

Explanation: Because all dynamic routes to the destination were removed, the IP layer attempted to add a saved replaceable static route to *dest/prefix_len* that uses *gateway* to the routing table. The add failed for the reason given.

dest/prefix_len is the IP address and prefix length of the route destination.

gateway is the IP address of the route gateway. *gateway* will be :: if directly connected.

reason is the reason the add failed and is one of the following:

1 The gateway that the route uses is unreachable.

2 The interface that the route uses is not valid.

3 The route was defined incorrectly.

System Action: The failing route is not added to the TCP/IP routing table, and processing continues. Other defined replaceable static routes might be added to the TCP/IP routing table. The definition of the failed replaceable static route is retained and TCP/IP might attempt to add it again, if there are no other ways to reach the destination.

User or Operator Response: Contact the system programmer.

System Programmer Response: Response depends on reason code:

1 This might be a temporary condition that can resolve itself when the gateway becomes reachable. If network actions are needed to make the gateway reachable, perform them (for example bringing up routers or connections). Otherwise, check the ROUTE statement for the replaceable static route, and if the gateway field is incorrect, fix it.

2 Verify the interface name used on the ROUTE statement of the replaceable static route.

3 Correct invalid syntax on the ROUTE statement of the replaceable static route.

Source Data Set: IPV6RTE

Procedure Name: InstallReplRoutes_6

EZZ4350I AN EXCESSIVE NUMBER OF HALF-OPEN CONNECTIONS ARE QUEUED FOR LISTENING PORT: *listen port*

Explanation: An excessive number of half-open connections have been queued for the specified listening port. Many half-open connection are created when connection requests are received from a remote host but that remote host does not acknowledge the connection acknowledgments sent by TCPIP in a timely fashion. It is possible that this condition is caused by a denial of service attack on that port.

listen port is the Listening Port Number.

System Action: TCP/IP will automatically implement a connection acceptance algorithm to minimize the impact of these incomplete connection requests.

User or Operator Response: Contact your system programmer.

System Programmer Response: Contact your security administrator about this possible denial of service attack.

Source Data Set: EZBTCRDE

EZZ4360I *jobname* **ECSA CONSTRAINED**

Explanation: This message is issued when ECSA storage usage is at or higher than 80% of the ECSALIMIT parameter on the GLOBALCONFIG statement. TCP/IP removes the message when storage usage drops below 75% of the ECSALIMIT.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues. Subsequent storage requests might fail.

User or Operator Response: Issue DISPLAY TCPIP,,STOR command to determine current storage usage and

limits. Save the system log and request a dump for problem determination. See the *z/OS Communications Server: IP Diagnosis* for more information.

System Programmer Response: Verify that the ECSALIMIT value is correct. If necessary, increase the limit on GLOBALCONFIG with a VARY TCPIP,,OBEYFILE command.

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4361I *jobname* **ECSA CRITICAL**

Explanation: This message is issued when ECSA storage usage is at or higher than 90% of the ECSALIMIT parameter on the GLOBALCONFIG statement. TCP/IP removes the message when storage usage drops below 85% of the ECSALIMIT.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues. Subsequent storage requests might fail.

User or Operator Response: Issue DISPLAY TCPIP,,STOR command to determine current storage usage and limits. Save the system log and request a dump for problem determination. See the *z/OS Communications Server: IP Diagnosis* for more information.

System Programmer Response: Verify that the ECSALIMIT value is correct. If necessary, increase the limit on GLOBALCONFIG with a VARY TCPIP,,OBEYFILE command.

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4362I *jobname* **ECSA EXHAUSTED**

Explanation: This message is issued when an ECSA storage allocation fails, or the ECSA storage usage reaches 98% of the ECSALIMIT parameter on the GLOBALCONFIG statement.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues. Subsequent storage requests might fail.

User or Operator Response: Issue DISPLAY TCPIP,,STOR command to determine current storage usage and limits. Save the system log and request a dump for problem determination. See the *z/OS Communications Server: IP Diagnosis* for more information.

System Programmer Response: Verify that the ECSALIMIT value is correct. If necessary, increase the limit on GLOBALCONFIG with a VARY TCPIP,,OBEYFILE command. If system ECSA is too small, update system parmlib member IEASYSxx. See the *z/OS MVS Initialization and Tuning Reference* for information about the CSA parameter.

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4363I *jobname* **ECSA SHORTAGE RELIEVED**

Explanation: This message is issued when a previous ECSA storage shortage is resolved.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues.

User or Operator Response: None

System Programmer Response: None

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4364I *jobname* **POOL CONSTRAINED**

Explanation: This message is issued when TCP/IP pooled private storage usage is at or higher than 80% of the POOLLIMIT parameter on the GLOBALCONFIG statement. TCP/IP removes the message when storage usage drops below 75% of the POOLLIMIT.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues. Subsequent storage requests might fail.

User or Operator Response: Issue DISPLAY TCPIP,,STOR command to determine current storage usage and limits. Save the system log and request a dump for problem determination. See the *z/OS Communications Server: IP Diagnosis* for more information.

System Programmer Response: Verify that the POOLLIMIT value is correct. If necessary, increase the limit on GLOBALCONFIG with a VARY TCPIP,,OBEYFILE command.

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4365I *jobname* **POOL CRITICAL**

Explanation: This message is issued when TCP/IP pooled private storage usage is at or higher than 90% of the POOLLIMIT parameter on the GLOBALCONFIG statement. TCP/IP removes the message when storage usage drops below 85% of the POOLLIMIT.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues. Subsequent storage requests might fail.

User or Operator Response: Issue DISPLAY TCPIP,,STOR command to determine current storage usage and limits. Save the system log and request a dump for problem determination. See the *z/OS Communications Server: IP Diagnosis* for more information.

System Programmer Response: Verify that the POOLLIMIT value is correct. If necessary, increase the limit on GLOBALCONFIG with a VARY TCPIP,,OBEYFILE command.

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4366I *jobname* **POOL EXHAUSTED**

Explanation: This message is issued when a TCP/IP pooled private storage allocation fails, or the pooled private storage usage reaches 98% of the POOLLIMIT parameter on the GLOBALCONFIG statement.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues. Subsequent storage requests might fail.

User or Operator Response: Issue DISPLAY TCPIP,,STOR command to determine current storage usage and limits. Save the system log and request a dump for problem determination. See the *z/OS Communications Server: IP Diagnosis* for more information.

System Programmer Response: Verify that the POOLLIMIT value is correct. If necessary, increase the limit on GLOBALCONFIG with a VARY TCPIP,,OBEYFILE command.

Source Data Set: TCPIP

Procedure Name: EZBITSCU

EZZ4367I *jobname* **POOL SHORTAGE RELIEVED**

Explanation: This message is issued when a previous TCP/IP pooled private storage shortage is resolved.

jobname is the jobname associated with the procedure used to start TCP/IP.

System Action: Processing continues.

User or Operator Response: None

System Programmer Response: None

Source Data Set: TCP/IP

Procedure Name: EZBITSCU

EZZ4820I Modify command ignored, incorrect parm(s): *parms*

Explanation: Incorrect parameters were passed to OROUTED from a MODIFY command.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters from the MODIFY command line.

Source Data Set: EZBRDMAI

Procedure Name: do_modify, parse_parms

EZZ4823I Tracing debug packets *action timestamp*

Explanation: Debug packets tracing is enabled or disabled. The packets are displayed in data format.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: modifydebuglevel

EZZ4824I Modify command ignored, trace levels exceeded

Explanation: An incorrect number of trace levels (-t's) were passed from a MODIFY command.

System Action: OE RouteD continues.

User or Operator Response: None.

System Programmer Response: Specify a correct number of -t's in the MODIFY command line parameters.

Source Data Set: EZBRDMAI

Procedure Name: parse_parms

EZZ4825E *name***/udp:**

Explanation: A UDP port number for the router service *name* was not assigned in the SERVICES file or data set.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: The following search order is used to find the services data set or file:

1. /etc/services
2. *userid*.ETC.SERVICES
3. *hlq*.ETC.SERVICES

Verify that the SERVICES file or data set has an entry of the form: *name* port/udp. The entry must start in column one and entered in lowercase. Also, verify that port has been reserved for OROUTED under the PORT statement in the *hlq*.PROFILE.TCPIP data set.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4826E Terminating since clients require a socket

Explanation: OROUTED attempted to open a socket on a well known or user-defined port and service name but the open was not successful, or the socket could not be bound to an internet address and port number. Other routers will not be able to communicate with OROUTED because a socket is not available.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Examine previous messages to determine the nature of the error as indicated by a detailed socket error message. Correct the problem as indicated by the error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4827E Usage: orouted ((–t)(–t)(– t)(–t)) (–sdl –sll–svl–sdvl–svdl –stl–svh) (–q) (–g)(–dp)(–ep)(–d) (–h)(–hv) (–del)(–c)

Explanation: OROUTED was passed parameters that are not supported or were not formatted correctly.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that the parameters are correct and are separated by spaces. See the *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4828I Input parameter(s): *parameter*

Explanation: This message displays the input parameters of the OROUTED server as entered by the user.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: parse_parms

EZZ4829I Waiting for incoming packets

Explanation: OROUTED is waiting for datagrams to arrive from other routers. Each time that OROUTED finishes processing an event such as an incoming datagram or a timer that expires, OROUTED issues this message and waits for the next event. These messages should occur at least every 15 seconds.

System Action: Processing is suspended until a timer expires or a datagram arrives from another router.

User or Operator Response: None.

System Programmer Response: If 6 or more of these messages occur consecutively, OROUTED is not receiving any datagrams from other routers. Verify that another router is active on a directly connected network. If so, examine the status of the appropriate network interface.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4830I The main select was interrupted

Explanation: An error occurred while OROUTED was waiting for an event to occur. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Verify that TCP/IP is running.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4831I Send delayed dynamic update

Explanation: A routing update that was delayed to prevent packet storms was transmitted. This occurs 2–5 seconds after a dynamic update was issued.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4832I While receiving a packet

Explanation: An error occurred while attempting to receive a packet from a client. The incoming packet is discarded. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Refer to the next error message and correct the error.

Source Data Set: EZBRDMAI

Procedure Name: process

EZZ4833E A socket could not be created

Explanation: OROUTED could not open a new socket. A subsequent message will be issued with more error information.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that TCP/IP is active, that another program is not using the OROUTED's port, that the well known or user-defined port was reserved in the PORT statement of *hlq*.PROFILE.TCPIP data set, and that the well known or user-defined port and service name have been specified in the SERVICES data set or file. Correct the problem as indicated by the socket error message. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZBRDMAI

Procedure Name: getsocket

EZZ4834E Broadcasting cannot be enabled on the socket

Explanation: OROUTED cannot enable the socket for broadcasting. OROUTED must be able to broadcast over interfaces that support broadcasting to communicate with other routers.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDMAI

Procedure Name: getsocket

EZZ4835E The socket bind failed

Explanation: OROUTED was unable to associate an internet address and port number with the newly created socket. This might be because another application is using the port.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that no other application is using the router port. Use the onetstat -a command and look for a port in the Local Socket column that matches the router port in the SERVICES file or data set. You should reserve this port for OROUTED's exclusive use by adding an entry to the PORT statement in the *hlq*.PROFILE.TCPIP data set.

Source Data Set: EZBRDMAI

Procedure Name: getsocket

EZZ4836I *function*: **socket create failed**

Explanation: OROUTED was unable to obtain a socket for temporary use by *function* while initializing interfaces or obtaining interface information from TCP/IP's interface tables. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: various

Procedure Name: *function*

EZZ4837I ioctl (get interface configuration)

Explanation: OROUTED encountered an error while attempting to obtain the network interface configuration. A subsequent message will be issued with more error information. The most likely error is ENOBUFS, which indicates an internal TCP/IP problem. The interface in error is skipped.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4838I *interface*: **ioctl (get interface flags)**

Explanation: OROUTED encountered an error while obtaining the interface flags for *interface*. A more specific message follows, which indicates the error returned from the ioctl. The interface is skipped.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: If the error is ENOENT, verify that a BSDROUTINGPARMS statement is in the *hlq*.PROFILE.TCPIP data set. Otherwise, contact your IBM Software Support Center. For more information, see the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4839I *interface*: **ioctl (get destination address)**

Explanation: OROUTED encountered an error while attempting to obtain the destination address from the point-to-point interface *interface*. A subsequent message will be issued with more error information. The interface in error is skipped.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4840I *interface*: **ioctl (get broadcast address)**

Explanation: OROUTED encountered an error while attempting to obtain the broadcast address for interface *interface*. A subsequent message will be issued with more error information. The interface is skipped.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4841I *interface*: **ioctl (get metric)**

Explanation: OROUTED encountered an error while obtaining the metric for interface *interface*. A subsequent message will be issued with more error information. The metric for the interface is set to zero.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4842I **Ignoring incorrect metric on interface** *interface*

Explanation: An incorrect metric value was specified for this interface in the BSDROUTINGPARMS statement. The default metric of 0 (zero) is used.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Specify a correct metric value in the range from 0 to 14.

Source Data Set: EZBRDINI

Procedure Name:

EZZ4843I *interface*: **ioctl (get netmask)**

Explanation: OROUTED encountered an error while obtaining the subnet mask for *interface*. A subsequent message will be issued with more error information. The interface in error is skipped.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: ifinit

EZZ4844I **Function** *function* **out of memory**

Explanation: OROUTED is unable to allocate the free storage needed to process an interface.

System Action: OROUTED continues.

User or Operator Response: None.

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

Source Data Set: EZBRDINI, EZBRDRMT

Procedure Name: ifinit, rtadd

EZZ4845I *entry*

Explanation: A warning is about to be issued for the indicated entry in the GATEWAYS file or data set. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4846I *entry*

Explanation: An error is about to be issued for the indicated entry in the GATEWAYS file or data set. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4847I **Second element ignored, changing to 'active'**

Explanation: The second element in the active gateway entry is detected to be not valid. This suggests that a configuration error might have occurred. OROUTED will assume the gateway entry to be an active gateway. The second element is changed to active.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4848I Gateway type '*type*' incorrect for active gateway

Explanation: An active GATEWAYS entry has a route type other than "active". GATEWAYS entries classified as active must have a route type of "active". The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4849I OE RouteD Server started

Explanation: OROUTED (OE RouteD) completed initialization of network interfaces with direct routes and is ready to process RIP packets for IP dynamic routing. Any routes defined in the GATEWAYS file or data set have also been processed.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4850I Processing interface *interface*

Explanation: The indicated interface was found in TCP/IP's interface table and is being added to OROUTED's interface table.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4851I First two elements must be 'active' for active gateway

Explanation: A GATEWAYS entry for the gateway definition has a route type of "active", but the first 2 elements are not defined as active, as required for an active gateway definition. The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4852I Incorrect gateway type '*type*'

Explanation: The GATEWAYS entry for the gateway definition has a gateway type that is not valid. Valid gateway types are: active, external, or passive. The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4853I packet from unsupported address family *family*, **cmd** *command*.

Explanation: An RIP packet was received by OROUTED from a foreign router in an unsupported address family. Only the internet address family is supported. The packet is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Ignore the error or locate the foreign router that originated the unsupported RIP packet and have it stopped.

To help locate the foreign router, turn on tracing (-t -t). The incoming RIP packets will display the source address and port. If you need to identify the interface which the RIP packet was received over, use a packet trace with filtering on source and destination ports 520.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4854I RIP version 0 packet received from *router*! **(cmd** *command***).**

Explanation: An RIP Version 0 packet was received from the specified router. This version of RIP is obsolete and not supported by OROUTED. The Version 0 packet is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the router that is sending the version 0 packets.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4855I trace command from unknown router *router*

Explanation: A TRACEON or TRACEOFF packet was received from a router that is either not directly attached, is reached using a passive interface, or reached using an interface that cannot handle broadcast or point-to-point traffic. The packet is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: If the originating router directed the packet using a passive link, an indirect path, or an interface that cannot handle broadcast or point-to-point traffic, then correct the originating router. Otherwise contact your IBM Software Support Center.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4856I Incorrect packet from passive interface *interface*

Explanation: OROUTED received a response packet from a local passive interface *interface*. Although OROUTED receives routing updates from a client for passive interfaces, it cannot produce the routing updates. The packet is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate the router that is producing the incorrect response packet and correct the problem.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4857I Packet from unknown router *ipaddr* **(***reason***)**

Explanation: A Routing Information Protocol (RIP) response was received from a router that is not directly connected by a broadcast network, point-to-point (NCST) network, or an active gateway as defined in GATEWAYS file or data set. The *reason* is one of the following:

Interface in strange state	The network does not support broadcast or point-to-point transmissions.
Iflookup failed	Not directly connected.
this Link is PASSIVE!	Cannot update.

The packet is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the router that originated the packet.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4858I route from *router* **in unsupported address family** *family*

Explanation: An incoming route from another router is in an address family that is not supported by OROUTED. Only internet addresses are supported. The route is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the router that originated the route.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4859I Illegal address *host* **in route from** *router*

Explanation: An internet address that is not valid was received in an update from router *router*. A previous message indicates the nature of the problem with the address. The route that is not valid is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the *router* that originated this route.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4860I bad metric *metric* **in route to** *destination* **from** *router***.**

Explanation: A route was received from *router* that contained a metric that was not in the range 1–16. The route is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate and correct the *router* that originated this route.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4861I Send dynamic update

Explanation: Changes to the routing tables have occurred and enough time has passed since the last update so OROUTED is allowed to update adjacent routers immediately rather than waiting until the next broadcast timer expires. Dynamic updates are not made if they occur too close to another update because network performance might be affected, and they are not made if another dynamic update is pending.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4862I Delay dynamic update

Explanation: Changes have been made to OROUTED's routing tables, but a dynamic update cannot be made because another update occurred recently, or another dynamic update is already pending. When the last update was made, a timer was set for 2–5 seconds from that time. When this timer expires, all pending dynamic updates are sent.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4863I Inhibit dynamic update for *number* **seconds**

Explanation: A dynamic update was sent, further dynamic updates are not sent during the next 2–5 seconds.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4864I inet_output: sendto destination *destination*

Explanation: An error occurred while attempting to transmit a packet. The packet is not sent. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDAF

Procedure Name: inet_sendto

EZZ4865I Trace buffers not initialized for interface *interface*

Explanation: One or both of the trace buffers for the specified interface could not be initialized. Tracing is limited.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Increase the region size for OROUTED and restart.

Source Data Set: EZBRDTRC

Procedure Name: traceinit

EZZ4866I Tracing *action timestamp*

Explanation: Tracing is enabled or disabled.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: various

EZZ4867I Displaying internal *type* **table**

Explanation: OROUTED's internal IP routing or interface table is displayed for diagnosis.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDRMT, EZBRDIF

Procedure Name: dsp_rrtables, dsp_iftables

EZZ4868I Tracing actions *action timestamp*

Explanation: The current tracing level of actions was either enabled or disabled. If enabled, messages will be issued for actions such as adding, changing, or deleting a route. Additional messages for actions such as waiting for incoming packets and dynamic updates are also issued.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4869I Tracing packets *action timestamp*

Explanation: The current tracing level of packets was enabled or disabled. If enabled, messages will be issued for packets sent and received, in addition to the output displayed for the actions level.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4870I Tracing history *action timestamp*

Explanation: The current tracing level of history was enabled or disabled. If enabled, messages will be issued for history tracing data on a per-interface basis, in addition to output displayed for the packets level. The history tracing data is displayed whenever an interface becomes inactive. It shows the latest traces of actions, packets, and packet contents before the interface became inactive.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4871I Tracing packet contents *action timestamp*

Explanation: The current tracing level of packet contents was enabled or disabled. If enabled, messages will be issued which displays the contents of packets sent or received, in addition to output displayed at lower tracing levels. Additional messages such as request for full routing tables and unknown address family in routing information are also issued.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: bumploglevel

EZZ4882I *timestamp***:**

Explanation: A full time stamp is issued showing date and time so that traces that exceed one calendar day can be interpreted correctly.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: traceaction

EZZ4883I *action* **destination** *destination***, router** *router***, metric** *metric* **flags** *flags* **state** *state* **timer** *timer*

Explanation: A route to *destination* is added, changed, or deleted depending on the value of *action*. The following values for *action* are allowed:

ADD: A route to *destination* is added through *router* at a metric of *metric*.

CHANGE FROM:
The route to *destination* is changed, the old values are displayed.

CHANGE TO: The route to *destination* is changed, the new values are displayed.

DELETE: The route to *destination* is deleted.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: trace

EZZ4884I rtadd: incorrect address family

Explanation: An attempt is being made to add a route which is to a destination in an unsupported address family. Currently, OROUTED only supports routes to Internet addresses.

System Action: OROUTED continues, the route is not added.

User or Operator Response: None.

System Programmer Response: Identify the router which generated this route and correct the problem.

Source Data Set: EZBRDRMT

Procedure Name: rtadd

EZZ4885I * Variable subnetting not supported in TCPIP

Explanation: OROUTED was configured to use variable subnetting (RIP version 2), but the current TCP/IP stack is not configured to support variable subnetting.

System Action: OROUTED continues

User or Operator Response: Either reconfigure OROUTED to use only RIP version 1 or configure the TCP/IP stack to support variable subnetting. Variable subnetting can be enabled in the stack with the statement IPCONFIG VARSUBNETTING.

System Programmer Response: none

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4888I Interface *interface* **is passive**

Explanation: The *interface* is in a passive state, meaning that RIP traffic is disabled for the interface. Routing updates will not be broadcast to the interface and incoming routing updates are ignored. This may be the result of a RIP I/O filter option specified in the GATEWAYS file or data set. See the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4889I CHANGE metric destination *destination***, router** *router***, from** *old metric* **to** *new metric*

Explanation: The metric for the route to *destination* is changed from *old metric* to *new metric*. OROUTED will change the metric if it received a RIP packet from router *router* that has a different metric than the one in OROUTED's tables. The message is also issued when a route times out, and OROUTED changes the metric value to 16 (unreachable).

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: tracenewmetric

EZZ4890I toall: requested to skip interface *interface*

Explanation: The interface *interface* is skipped because the interface already received notification of a routing change. If a broadcast of the routing table was not sent recently and a routing change occurred, a dynamic routing update will be broadcast to other interfaces to inform adjacent routers of the routing change.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4891I * Packet history for interface** *interface* *******

Explanation: Tracing is set at the history level and the history trace data for the inactive interface is displayed.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumpif

EZZ4892I * End packet history *****

Explanation: Tracing is set at the packet history level and this message ends the history trace data for the inactivated message.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumpif

EZZ4897I *direction***: no packets.**

Explanation: Either the input or output trace buffer, depending on the value of *direction*, is empty.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumptrace

EZZ4898I *direction* **trace:**

Explanation: Tracing is set at the history level and either the input or output trace buffer is about to be displayed, depending on the value of *direction*.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumptrace

EZZ4899I *RIPcmd direction router* **->** *port* **ver** *version timestamp*

Explanation: An RIP datagram was received from or is about to be transmitted to another router, depending on the value of *direction*. A value of *to* indicates an outbound datagram, while a value of *from* indicates an inbound datagram. The version of the datagram is determined by the *version*. The type of the datagram is determined by the value of *RIPcmd* and one of the following:

REQUEST Indicates that the routing table information is requested. Individual routes or a complete routing table may be requested.

RESPONSE Indicates that the routing table information is broadcast.

TRACEON Request to start tracing to a specified data set.

TRACEOFF Request to stop tracing and close the current tracing data set.

The value of *port* is either the port number on which the datagram came in or goes out on, or in the case of 0, it indicates that an outbound datagram will go out on the port assigned to *router* in the SERVICES data set.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4900I **Bad cmd** *hex direction router* **->** *port timestamp*

Explanation: A malformed packet was encountered during tracing. The source of the packet is either *router* if *direction* is *from*, or OROUTED if *direction* is *to*.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: If the source of the packet is *router*, correct this router. Otherwise, contact your IBM Software Support Center.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4901I **(truncated record, len** *len***)**

Explanation: An RIP RESPONSE datagram was received that did not end on a route boundary. Either the packet extends beyond the last route, or it was truncated.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Verify that the incoming packet was built correctly and can be processed by other routers. Contact your IBM Software Support Center if the problem appears to be with OROUTED.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4902I **destination** *destination* **metric** *metric*

Explanation: A route to *destination* with the indicated metric is displayed by another router or OROUTED, depending on the contents of the last EZZ4899I message. A route is never advertised over the interface from which it was received.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4903I request for full tables

Explanation: A request for a complete routing table was sent or received depending on the contents of the last EZZ4899I message. This message is sent by OROUTED at startup over each interface so that OROUTED can obtain routing tables from all adjacent routers.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4904I Unknown address family (*family*) metric *metric*

Explanation: A route was found in the trace with an unknown address family.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: If the route originated from another router, locate that router and correct it. If the route originated from OROUTED, contact your IBM Software Support Center.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4905I TraceOn file = *dataset*

Explanation: A TRACEON packet was received, trace data is requested to go to *data set*.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTRC

Procedure Name: dumppacket

EZZ4909I Error adding route to *destination* through router *router*, metric *metric*

Explanation: An error occurred while attempting to add a route to *destination*. A subsequent message will be issued with more error information.

System Action: OROUTED continues, the route is not added.

User or Operator Response: None.

System Programmer Response: See EZZ4910E.

Source Data Set: EZBRDRMT

Procedure Name: tcp_add

EZZ4910I *function*: **SIOCADDRT ioctl failed**

Explanation: OROUTED was unsuccessful in adding a new route. The error occurred in the indicated function and is one of the following:

rtadd OROUTED attempted to add a new route to the TCP/IP IP routing table.

rtchange
OROUTED attempted change the metric of a route in TCP/IP's IP routing table by deleting the old route and adding a new one with the updated metric.

A subsequent message will be issued with more error information. Most likely the error is the result of a bad GATEWAYS entry, and common errors are:

EACESS
OROUTED is not authorized to add routes to TCP/IP's IP routing table.

EEXIST
The route already exists in TCP/IP's tables either because of a GATEWAY statement, a previous job execution of OROUTED, or lack of success in deleting the route.

ENETUNREACH
TCP/IP does not have a route to the router (nexthop) for this route. Routes in the GATEWAYS file or data set must have reachable routers.

ENOENT
TCP/IP did not have a link interface defined for the route to be added to.

EINVAL
An incorrect parameter was passed to TCP/IP.

System Action: The route is not added. OROUTED continues.

User or Operator Response: None.

System Programmer Response: These errors require the following responses:

EACESS
OROUTED can only be started from a RACF-authorized user ID. Verify that OROUTED is properly configured with RACF.

EEXIST:
Clear the IP routing table before restarting OROUTED. A VARY TCPIP,,OBEYFILE command with the word *GATEWAY* removes all routes.

ENETUNREACH:
Correct the GATEWAYS file or data set, either add a route to the router, or select another router for this route.

ENOENT:
Define the link interface on the BSDROUTINGPARMS statement in the *hlq*.PROFILE.TCPIP. A VARY TCPIP,,OBEYFILE command with the BSDROUTINGPARMS definitions may be used to add the link interfaces to TCP/IP.

EINVAL:
Contact your IBM Software Support Center.

Source Data Set: EZBRDRMT

Procedure Name: tcp_add

EZZ4912E **Error deleting route to** *destination* **through router** *router*, **metric** *metric*

Explanation: An error occurred while attempting to delete a route to *destination*. A subsequent message will be issued with more error information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Refer to the explanations for the specific error messages displayed with this one.

Source Data Set: EZBRDRMT

Procedure Name: tcp_del

EZZ4916I *action* **route to interface** *interface* **(timed out)**

Explanation: The route to *interface* is changed or deleted depending on the value of *action*. The value *changing* indicates that *interface* will be deleted then re-added. The value *deleting* indicates that the route to *interface*'s network or subnet will be deleted. The interface remains in OROUTED's interface list even though the route is deleted.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDRMT

Procedure Name: rtchange

EZZ4918I *function*: **SIOCDELRT ioctl failed**

Explanation: OROUTED was unsuccessful in deleting a route. The error occurred in *function*, and is one of the following:

rtdelete
OROUTED attempted to delete a route from TCPIP's routing table.

rtchange
OROUTED needed to change a route, but TCPIP does not allow routes to be changed explicitly. Instead, OROUTED attempted to delete the old route so that a new route can be added with the updated values.

rtdeleteall
OROUTED attempted to delete all routes from TCPIP's routing IP routing table.

A subsequent message will be issued with more error information. The most likely error is ESRCH, which indicates that the route that is to be deleted does not exist in TCPIP's routing tables.

System Action: OROUTED continues, the route is not deleted.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDRMT

Procedure Name: tcp_del

EZZ4919I **Using backup interface** *interface*

Explanation: A primary interface became inactive and a secondary *interface* is serving as the backup.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: delrouteforif

EZZ4920I **Add** *type* **route ignored, route exists on interface** *interface*

Explanation: A network, subnetwork, or destination type route already exist on *interface* that is active. As a general rule, no more than one route can be defined to a single destination and a route is added to the first available interface in the (sub)network according to the order in the HOME statement of *hlq*.PROFILE.TCPIP.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4921I Deleting route to interface *interface***? (timed out?)**

Explanation: An interface route is deleted even though the route exists in TCP/IP's tables. One possible cause is that the route was deleted due to a timed out situation. Another reason for the deletion is defining a passive route in the GATEWAYS file or data set, when the route was also defined in the BSDROUTINGPARMS or HOME section.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDRMT

Procedure Name: rtdelete

EZZ4922I Option(s): *options*

Explanation: Additional OROUTED options, specified in a GATEWAYS file or data set, are being processed. See the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4923E Incorrect option: *option*

Explanation: In the line entry for the GATEWAYS file or data set, the options definition has an incorrect option. Although other options may be processed normally, the incorrect option is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set. See the *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4924I Re-installing interface *interface*

Explanation: The previously deleted interface *interface* is reinstalled, because traffic was detected over this interface.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4925I Start of GATEWAYS processing:

Explanation: The GATEWAYS file or data set member is about to be processed. Processing messages may follow for the entries in the file or data set.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4926I End of GATEWAYS processing

Explanation: Processing is completed for the GATEWAYS file or data set.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4927I Zero metric not allowed, changing to one

Explanation: In the line entry for the GATEWAYS file or data set, the metric has a value of 0, which is not allowed. The range of valid metric values is 1 through 15. The metric is changed to a default value of 1.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4928E Out of memory while processing GATEWAYS

Explanation: OROUTED did not have enough storage available to process the GATEWAYS file or data set, which contains the addresses of gateways on the network.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Increase the region size for OROUTED and restart OROUTED. If the problem persists, contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4929I Port *port* assigned to *name*

Explanation: OROUTED listens for traffic from other routers on the specified port number *port* assigned to router service *name*.

System Action: TCP/IP continues. OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4930I Packet from router *router* ignored (*action*)

Explanation: OROUTED was configured to ignore packets from router *router*. Action *action* was applied to the packets.

System Action: OROUTED continues.

User or Operator Response: If it is desired to use packets from the router specified, update the gateways data set or file to accept packets from this router.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4931I Ignoring route *destination*, supernetting not supported

Explanation: The route to *destination* received in the RIP packet happens to be a supernet type of route and is ignored since the supernetting feature is not supported in TCP/IP. A supernet route is one where its subnet mask is less than the route's network class mask.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Verify that the VARSUBNETTING option is coded in the ASSORTEDPARMS or IPCONFIG statement in the PROFILE.TCPIP data set.

Source Data Set: None.

Procedure Name: None.

EZZ4932I ******** **

Explanation: Two of these banners enclose a message that might need attention when viewing the output.

System Action: TCP/IP continues. OROUTED continues.

User or Operator Response: None.

System Programmer Response: Read the enclosed message and, when necessary, resolve the indicated error.

Source Data Set: various

Procedure Name: various

EZZ4933I Unable to open a GATEWAYS *file/dataset* (*filename*)

Explanation: A gateways file or data set was not specified, or could not be opened. The indicated file or data set is the file or data set that OROUTED attempted to open.

System Action: TCP/IP continues. OROUTED continues.

User or Operator Response: None.

System Programmer Response: The OROUTED server uses the following search order to locate the GATEWAYS configuration data set or file:

1. If the environment variable GATEWAYS_FILE was defined, .OROUTED uses this value as the name of an MVS data set or HFS file to access the gateways file. The syntax for an MVS data set name is *"//'mvs.dataset.name'"*. The syntax for an HFS file name is *"/dir/subdir/file.name"*.
2. /etc/gateways
3. *hlq*.ETC.GATEWAYS

   Note: Only the first file in the search order that can be opened will be read to determine the gateway statements.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4934I Opening GATEWAYS *file/dataset* **(***filename***)**

Explanation: The GATEWAYS file or data set is being opened and the entries in the file or data set are read in for input.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4935E Incorrect parameter: *parameter*

Explanation: An incorrect parameter was passed to OROUTED. The parameter could either be passed from the command line parameters or from the default parameter list in the start procedure JCL.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the parameter from the command line parameters or in the default parameter list of the start procedure JCL.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4936I Adding *type route* **route** *destination* **via gateway** *gateway***, metric** *metric*

Explanation: The indicated route, defined in the GATEWAYS file or data set, is being added to OROUTED's routing table. The route to the gateway will not be replaced by a competing RIP route.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4937I Subnetwork mask unknown for *destination***, using network route** *route*

Explanation: The indicated subnetwork route, defined in the GATEWAYS file or data set, was explicitly coded as a "net" route type. Because the subnetwork mask for the destination subnetwork is unknown, OROUTED replaces the subnetwork route with a network route. OROUTED currently does not support variable subnetting.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDRMT

Procedure Name: rtadd

EZZ4938I Adding active gateway *ip_addr*, **metric** *metric*

Explanation: An active gateway *ip_addr* with metric *metric*, defined in the GATEWAYS file or data set, is being added to OROUTED's routing table. The route to the active gateway will be treated as a network interface.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4939E Trace levels exceeded maximum of 4 -t's

Explanation: The maximum number of -t's that are part of a valid request is four (-t-t-t-t). Each -t indicates an additional level of tracing.

System Action: OROUTED ends abnormally.

User or Operator Response: Specify a valid -t parameter for the trace request you would like. *z/OS Communications Server: IP Configuration Reference* describes the OROUTED parameters.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4940I Point-to-point interface, using *addrtype*

Explanation: The new interface is point-to-point and the destination address type, *addrtype*, can be either *dstaddr* or *broadaddr*. The address type is determined by the *destaddr* field definition for the interface in the BSDROUTINGPARMS statement of the TCP/IP profile. The destination address may be coded as zero or non-zero. If zero, the network or subnetwork directed broadcast address is used; otherwise, the unicast or host address representing the other end of a point-to-point link is used. This address is used for sending the routing information over an interface to the destination router or host. The exception is when the interface is multicast-capable, the multicast address is used.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4941I Interface *interface* **skipped, interface is multicast-incapable**

Explanation: The interface *interface* is skipped because OROUTED was configured with the RIP2 multicast option for the interface, but TCP/IP reported that the interface does not support multicast operation.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: When permitted, use the RIP2M supply control option to allow broadcasting RIP Version 1 packets over multicast-incapable interfaces and multicasting Version 2 packets over the multicast-capable interfaces.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4942I Not an internal interface

Explanation: The new interface does not appear to be associated with a real device. Most likely the interface is a pseudo-interface, created as a result of an active gateway definition in the GATEWAYS file or data set.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4943I Adding *type* route for interface

Explanation: The route using the network, subnetwork, or host type is being added to the interface in OROUTED's routing table.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4944I Function *function*: address family is out of range. The address is *address*.

Explanation: The route to the destination address has an unsupported address family. OROUTED cannot determine the network based on the address, therefore it cannot determine an interface that serves the logical network.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Non-IP addresses are not supported in this version of OROUTED. Either ignore the message, or have the router that originated the route stop sending non-IP addresses to this router. Look back through the output to find the last ADD or CHANGE for this destination to obtain the IP address of the router.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4945I ifwithnet: compare with *value*

Explanation: The route's network address is being compared to one of OROUTED's interface entries for a network number match.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4946I ifwithnet: interface has bad address family

Explanation: One of OROUTED's interface entries address families is not valid. OROUTED cannot determine the network based on the address, and cannot determine an interface that serves the logical network. Valid address family is AF_INET for the internet domain.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Either ignore the message, or correct the interface with a correct address family. Change the address family to a valid value. If the socket address is in an internet addressing family, the addressing family can be defined in the IN.H header file. See the *z/OS Communications Server: IP Application Programming Interface Guide* for information about valid addressing families. If the interface cannot be corrected, contact your IBM Software Support Center.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4947I netmatch *ipaddr1* **and** *ipaddr2*

Explanation: A network number match was found for the first address in the message with one of OROUTED's interface entries that has the second address as its network address.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: if_ifwithnet

EZZ4948I This interface is not point-to-point

Explanation: The interface OROUTED is currently handling is not a point-to-point interface. OROUTED will create a route to the network for the interface.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: if_ifwithnet

EZZ4949I Interface *interface* **not up**

Explanation: OROUTED determined that the indicated interface is not active. No routes will be added unless the interface is activated or reactivated.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT, EZBRDINI

Procedure Name: toall, ifinit

EZZ4950I Interface *interface* **ignored**

Explanation: The indicated interface address family is not valid. The interface is ignored to prevent network or subnetwork routes from being added.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Either ignore the message, or correct the interface with a valid address family. A common cause of an interface with an incorrect address family is that the interface is undefined in HOME statement of *hlq*.PROFILE.TCPIP. This may be normal in the case where multiple interfaces are reserved for Offload processing but are not defined as primary interfaces (for example, the PRIMARYINTERFACE statement in *hlq*.PROFILE.TCPIP). If the interface cannot be corrected, contact your IBM Software Support Center.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4954I *timer_value* **minute timer expired for route to** *destination*

Explanation: No Routing Information Protocol (RIP) packets have been received from OROUTED for the route to the destination *destination* in the last *timer_value* minutes. It is assumed that the destination router is no longer active. Depending upon the *timer_value*, one of the following actions is taken:

3 The route will have its metric changed to infinity for the next 2 minutes. The metric change is necessary to alert adjacent routers that the route is to this destination is unreachable. If OROUTED receives any RIP packets from the destination router during the 2 minute interval, OROUTED will restore the route by changing the metric to a valid one based upon the received RIP packet.

5 The route will be deleted from OROUTED's routing table.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: These actions require the following responses:

3 If the route was broadcast for a while, and then suddenly stopped, look for an adapter problem, or a problem with the physical line.

5 Examine OROUTED trace output and determine when the broadcasting has stopped for the route to the destination router. If the route to *destination* was only broadcast once in response to OROUTED's request for full route tables, then the problem may be with the way RIP packets are broadcast.

In these cases, determine if OROUTED is receiving the RIP packets by obtaining a UPD packet trace. This trace can also confirm the lack of traffic for the route's interface. Contact the IBM support center for assistance.

Source Data Set: EZBRDTMR

Procedure Name: timer

EZZ4955I **Ignoring route** *destination*, **filtered out**

Explanation: The specified destination route is being filtered out from the routing information in a outgoing RIP broadcast or a received RIP update. The destination route matches one of the RIP input and output filters, defined in OROUTED gateways data set or file, and is being selected for removal from the routing information.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP, EZBRDOUT

Procedure Name: rip_input, supply

EZZ4956I **Deleting interface** *interface_name*

Explanation: OROUTED is deleting the interface from its interface tables. The interface is unusable, since it may have been dynamically deleted via the VARY TCPIP command for a new HOME list. See *z/OS Communications Server: IP Configuration Reference* for more information about the VARY TCPIP command.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: ifdelete

EZZ4957I *timer_value* **second timer expired (***reason***)**

Explanation: Depending upon the *timer_value* and *reason*, one of the following actions is taken:

Value	Reason	Action
30	broadcast	Every 30 seconds, a timer expires that indicates that OROUTED must broadcast its routing tables to adjacent routers. At this time, OROUTED will send RIP response packets for each of OROUTED's interfaces that allow broadcasting to the adjacent routers.
m	rescan kernel for interfaces	Every *m* seconds, a timer expires that indicates that OROUTED must rescan TCP/IP's interface table (kernel) for any activated, reactivated, or deactivated interfaces. At this time, OROUTED will add any new interfaces to its interface table, recognize interface order according to HOME and PRIMARYINTERFACE statements in *hlq*.PROFILE.TCPIP, or marks the interface as active or inactive. In addition, OROUTED will add routes for the activated or reactivated interfaces and if necessary, issue alerts (triggered updates) to adjacent routers for deactivated interfaces to attempt routing recovery.
n	poll interfaces for status	Every *n* seconds, a timer expires that indicates that OROUTED must poll TCP/IP's interface table for any activated, reactivated, or deactivated interfaces. OROUTED will mark the interface as active or inactive and if necessary, issue alerts (triggered updates) to adjacent routers for deactivated interfaces to attempt routing recovery.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDTMR

Procedure Name: timer

EZZ4958I **supply** *destination* **->** *port* **via** *interface*

Explanation: OROUTED is broadcasting a routing information protocol (RIP) datagram to the destination address. The destination address can be either a broadcast address or a host address. If the port is zero, the current OROUTED port will be used from the SERVICES file or data set. Otherwise, the datagram will be transmitted to the specified port over the interface to the destination.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: supply

EZZ4959I **Interface** *interface* **up**

Explanation: The specified interface, which was not previously active, is now active. The interface might have been inactive or re-installed.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4960I Incorrect host or (sub)network address '*destination*'

Explanation: In the line entry for the GATEWAYS file or data set, the gateway definition has an incorrect destination address. The destination must be either a resolvable host name, or an IP address in dotted decimal notation. The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4961I Incorrect gateway address '*gateway*'

Explanation: In the line entry for the GATEWAYS file or data set, the gateway definition has a gateway address that is not valid. The gateway address must be either a resolvable gateway name, or an IP address in dotted decimal notation. The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4962I Incorrect route type '*type*'

Explanation: In the line entry for the GATEWAYS file or data set, the gateway definition route type is not valid. Allowable route types are host, for host route, net, for network or subnetwork route, and active for a route to be treated as a network interface. The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4963I (no gateway definitions)

Explanation: The GATEWAYS file or data set does not contain any definitions for configuring the routing tables on your host.

System Action: TCP/IP continues.

User or Operator Response: Include appropriate definitions in the GATEWAYS file or data set for configuring the routing tables on your host before starting OROUTED.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4964I Incorrect metric, changing to one

Explanation: In the line entry for the GATEWAYS file or data set, the metric has an incorrect value. The metric is changed to 1 because the gateway internet address matches one of the home internet addresses on the originating host. The destination host or network address is one hop away.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4965I Virtual interface

Explanation: The new interface is a virtual and is used for fault tolerance support. The virtual interface always stays active and will never see a physical failure. Network and/or subnetwork virtual routes are being added to the interface.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4972I Incorrect *type* value: *value*

Explanation: In the line entry for the GATEWAYS file or data set, the options definition contains an incorrect value. Although other options may be processed normally, the incorrect option is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set. See the *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4973I Unknown interface name (*interface*) and/or address (*ip_addr*)

Explanation: In the line entry for the GATEWAYS file or data set, the options definition contains interface information that is not in OROUTED's interface table. Most likely, the interface name is misspelled or the interface's internet address is specified incorrectly. Although other options may be processed normally, the incorrect option is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file or data set. See the *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

EZZ4974I Issuing Kill Route Messages

Explanation: OROUTED detected an outage on a primary or alternate adapter and is issuing an alert 'Kill Route Messages' to adjacent routers over another interface providing the redundant path. This alert is used to notify adjacent routers to remove their routes to the failing adapter so that they can receive new routing information from the new interface and perform routing switches immediately.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINI

Procedure Name: addrouteforif

EZZ4980I Using catalog '*file*' for OE RouteD messages.

Explanation: The messages issued by OROUTED (OE RouteD) will be retrieved from the message catalog named *file*.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4981I Unable to open message catalog '*file*'.

Explanation: An attempt was made to open OROUTED's message catalog (named '*file*') in the directory determined by the NLSPATH and LANG environment variables, but the catalog could not be opened. A subsequent message will be issued with more error information.

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

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external catalog is desired, stop the server, correct the problem as indicated by the error text, restart the server. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. If the default messages are acceptable, no action is necessary.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4982I Using OE RouteD's default messages

Explanation: Because OROUTED (OE RouteD) was unable to open the message catalog, the default messages will be used.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4983I *errnum* **(ERRNO=***errno* **ERRNO2=***errnojr***)**

Explanation: After a system error, *errnum*, ERRNO and ERRNO2 are displayed

errno is the decimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the reason codes are listed.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Refer to the *z/OS UNIX System Services Messages and Codes* for more information about *errno* and *errnojr* error codes.

Source Data Set: EZBRDTRC

Procedure Name: sys_err

EZZ4984I Cannot determine the High Level Qualifier (HLQ). Using default HLQ of *hlq*

Explanation: Because OROUTED was unable to determine the HLQ, the default HLQ of *hlq* will used.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Check the OMVS resolver configuration file or data set to ensure that value of system variable DATASETPREFIX is set properly.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4985I Setting High Level Qualifier (HLQ) to '*hlq*'

Explanation: OROUTED set the HLQ variable to the system variable DATASETPREFIX.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4986I OE RouteD could not determine TCPIPjobname, using default of 'INET'

Explanation: OROUTED (OE RouteD) could not determine the jobname for the TCP/IP stack that it is to associate itself with. A default value of 'INET' will be used for TCPIPjobname.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for OROUTED (OE RouteD) to communicate with a particular stack, the TCPIPjobname should be set in the appropriate resolver configuration file or data set. See the *z/OS Communications Server: IP Configuration Guide* for information about the TCPIP.DATA search order.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4987I OE RouteD could not establish affinity with '*jobname*'

Explanation: OROUTED (OE RouteD) cannot communicate with the TCP/IP stack *jobname*. OROUTED (OE RouteD) attempted to use the OE Socket call setibmopt() to associate itself with the TCP/IP instance *jobname*. This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifier' format of the MVS start command was used) of the TCP/IP instance to which OROUTED (OE RouteD) is to be associated with. A subsequent message will be issued with the *errcode* and *reason* for the setibmopt() failure.

System Action: OROUTED (OE RouteD) ends abnormally.

User or Operator Response: None.

System Programmer Response: For OROUTED (OE RouteD) to communicate with a particular stack, the jobname (as determined by the system variable TCPIPjobname) must match "xxxxx" where "xxxxx" is set in the BPXPRMxx

member used to start OMVS. ″xxxxx″ is set in the SUBFILESYSTYPE NAME(xxxxx) for ENTRYPOINT(EZBPFINI).

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4988I OE RouteD established affinity with '*jobname***'**

Explanation: OROUTED (OE RouteD) successfully connected to the stack *jobname*.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4989I OE RouteD MODIFY command processed; see syslog output

Explanation: OROUTED (OE RouteD) processed a MODIFY command.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: do_modify

EZZ4990I OE RouteD server initializing. Level *level*

Explanation: OROUTED (OE RouteD) server is starting initialization. Internal OE RouteD level number is displayed.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4991I OE RouteD: sigaction() failed for *signal* **:** *reason*

Explanation: OROUTED (OE RouteD) encountered an error attempting to set up the signal handler for the signal specified by *signal*. *reason* is the error returned by the C runtime library for the failing sigaction() call. If the signal handler is not correctly enabled, OROUTED (OE RouteD) will continue processing, but certain functions controlled by the failing signal will not function properly. Functions controlled by the signals are:

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

SIGTERM
handler controls cleanup of resources during termination.

SIGPIPE
handler allows OROUTED (OE RouteD) to detect when the connection to OE Traceroute was terminated.

A subsequent message will be issued with more error information.

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

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ4992I OE RouteD: Abend detected

Explanation: This message indicates that an internal programming error caused OROUTED (OE RouteD) to abnormally terminate.

System Action: OROUTED (OE RouteD) ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: abndhand

EZZ4993I OE RouteD: Terminated

Explanation: OROUTED (OE RouteD) received the SIGTERM signal

System Action: OROUTED (OE RouteD) exits

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: termhand

EZZ4994I Incorrect routing filter mask '*mask*'

Explanation: In the line entry for OROUTED gateways file or data set, the gateway definition has an invalid route filter mask. The mask must be either a resolvable mask name, or a 32-bit value in dotted decimal notation. The gateways entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the gateways file or data set.

Source Data Set: EZBRDINI

Procedure Name: ParseOptions

EZZ4995I Interface *interface* skipped, supply is suppressed

Explanation: The broadcasting of routing information is being suppressed for the specified interface. For this reason, the interface is being skipped while OROUTED is supplying routing information to list of interfaces. An interface supply option, defined in the OROUTED gateways file or data set, determines whether or not OROUTED can supply routing information over an interface.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ4996I Ignore source VIPA option cannot be enabled on the socket

Explanation: An attempt was made to set the IBM socket option to ignore source VIPA addresses in outbound IP datagrams for RIP packets. VIPA addresses cannot be used as source IP addresses since adjacent routers rely on real (physical) IP addresses to determine gateway addresses for their routing tables; otherwise, adjacent routers will discard the RIP packets.

System Action: OROUTED ends abnormally.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: EZBRDMAI

Procedure Name: get_socket

EZZ4997I Unsupported RIP command *command* **received from** *source*

Explanation: OE Routed received an unsupported RIP command *command* from *source*.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate the router that is producing the unsupported RIP command and correct the problem.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ4998I Gateway address '*address***' not on a routing interface in the network**

Explanation: The indicated route, defined in OROUTED's gateways data set or file, referenced an unknown routing interface based upon the gateway address. The route definition is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the gateways data set or file. Verify that the gateway address is correct and a valid routing interface is defined.

Source Data Set: EZBRDINI

Procedure Name: gwkludge

Chapter 5. EZZ5*xxxx* messages

EZZ5000I Unable to open OE RouteD profile *profile*

Explanation: A profile data set or file was not specified or could not be opened. This message indicates which data set or file the open was attempted on.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: The OROUTED (OE RouteD) server uses the following search order to locate the routed profile configuration data set or file:

1. If the environment variable ROUTED_PROFILE was defined, OROUTED uses this value as the name of an MVS data set or HFS file to access the routed profile. The syntax for an MVS data set name is *"//'mvs.dataset.name'"*. The syntax for an HFS file name is *"/dir/subdir/file.name"*.
2. /etc/routed.profile
3. *hlq*.ROUTED.PROFILE

   Note: Only the first file in the search order that can be opened will be read to determine the routed profile.

Source Data Set: EZBRDMAI

Procedure Name: read_profile

EZZ5001I Opening OE RouteD profile *profile*

Explanation: The specified OROUTED (OE RouteD) profile data set or file is being opened. Entries in the data set or file are read in for input.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: read_profile

EZZ5002I *option value*

Explanation: The OROUTED profile data set or file was read and the *option* and its associated *value* are displayed.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: read_profile

EZZ5003I Unknown next hop address *ipaddress* **for route** *destination* **from router** *router*

Explanation: An unknown next hop address *ipaddress* was received in a RIP packet for a route to *destination* from *router*. The route is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP

Procedure Name: rip_input

© Copyright IBM Corp. 1994, 2002

EZZ5004I Address is experimental, or has non-zero port

Explanation: An incorrect internet address was encountered, which is either in an experimental address class, or is using an unusual port number (Routing Information Protocol (RIP) packets only). The address is being validated to ensure that a network user does not pretend to be a router in order to change the routing table of nearby routers. The route entry is discarded since the address is not considered as a valid destination address for a route.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Identify the machine or user which generated the packet in question and correct the problem.

Source Data Set: EZBRDAF

Procedure Name: inet_checkhost

EZZ5005I Incorrect internet address

Explanation: An internet address in an incoming route is determined not to be a member of any defined internet address class. The route is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate the router which originated the packet and correct the problem.

Source Data Set: EZBRDAF

Procedure Name: inet_checkhost

EZZ5006I A 'must-be-zero' field is non-zero

Explanation: An internet address in an incoming route contains a nonzero value in a field which must be zero. The incoming route is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Locate the router which originated the packet and correct the problem.

Source Data Set: EZBRDAF

Procedure Name: inet_checkhost

EZZ5007I Response from non-router *ipaddress*

Explanation: A Routing Information Protocol (RIP) response packet was received with an incorrect port number. All routers on a network must agree on the port number that is used to exchange routing information, and this port must be restricted so that other applications cannot generate routing updates. Either a router is configured using the wrong port number, or an application is issuing RIP routing updates. The RIP response is discarded.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Reconfigure the router to use a correct port number and locate the application which is generating the updates and correct the problem.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ5008I Unknown keyword (*keyword*) in OE RouteD profile

Explanation: An unknown keyword *keyword* was used in a statement in the OROUTED (OE RouteD) profile data set or file. The statement is ignored.

System Action: OROUTED (OE RouteD) continues.

User or Operator Response: None.

System Programmer Response: Correct the OROUTED (OE RouteD) profile data set or file

Source Data Set: EZBRDMAI

Procedure Name: read_profile

EZZ5009I Unknown RIP *rip_control* control value (*value*)

Explanation: The RIP control values (supply or receive), specified on the RIP_SUPPLY_CONTROL or RIP_RECEIVE_CONTROL entry in the OROUTED profile data set or file, or specified on the options statement entry for an interface, contains an incorrect value. In case of incorrect values, OROUTED will default the RIP supply control to *RIP1* and receive control to *ANY*.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the OROUTED profile or GATEWAYS data set or file by specifying a supported supply or receive control value.

Source Data Set: EZBRDMAI

Procedure Name: read_profile

EZZ5010I RIP2 authentication key incorrect

Explanation: The RIP Version 2 authentication key, specified on the RIP2_AUTHENTICATION_KEY entry in the OROUTED profile data set or file, or on the options statement entry in the GATEWAYS data set or file for an interface, is invalid. The authentication key may have contained unsupported characters or have exceeded the maximum 16 characters allowed or was not enclosed in quotes. The authentication key is ignored and no authentication check is performed.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the OROUTED profile or GATEWAYS data set or file.

Source Data Set: EZBRDMAI, EZBRDINI

Procedure Name: read_profile, ParseOptions

EZZ5011I *rip_control rip_version* packets on interface *interface* not allowed

Explanation: A RIP Version 1 or Version 2 packet is ignored depending upon the settings of the RIP supply or receive controls specified in the GATEWAYS data set or file for an interface or in the OROUTED profile data set. If there are no RIP control settings for an interface, OROUTED will use the one from the profile settings.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the OROUTED profile or GATEWAYS data set or file.

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ5012I RIP2 packet from router *router* not authorized

Explanation: A RIP Version 2 packet, received from *router*, is ignored due to an authentication key mismatch. Authentication is enabled for RIP Version 2 packets according to the interface options in GATEWAYS data set or file, or in the OROUTED profile data set or file. If there are no interface settings, OROUTED will use the one from the profile settings.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDINP,

Procedure Name: rip_input

EZZ5013I RIP2 authentication *action* at *level* level (*interface*)

Explanation: RIP Version 2 authentication is enabled or disabled at the specified level for an interface or for all interfaces.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI,EZBRDINI

Procedure Name: read_profile, ParseOptions

EZZ5014I Incorrect subnetwork mask '*mask*'

Explanation: In the line entry for GATEWAYS data set or file , the gateway definition has a subnetmask that is not valid. The subnetmask must be a bit mask in dotted-decimal notation. The bits must be contiguous in the network portion of the subnetmask The GATEWAYS entry is ignored.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS data set or file

Source Data Set: EZBRDINI

Procedure Name: ParseOptions

EZZ5015E OE RouteD initialization terminated because TCP/IP Base feature of OS/390 is not enabled.

Explanation: OROUTED (OE RouteD) issued a query for product=OS/390 feature=TCP/IP_BASE but IFAPRDxx parmlib member does not indicate that the feature TCP/IP_BASE is enabled. OROUTED (OE RouteD) will only run if TCP/IP_BASE is enabled.

System Action: OROUTED (OE RouteD) initialization is terminated.

User or Operator Response: Contact the system programmer.

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

Source Data Set: EZRDMAI

Procedure Name: main

EZZ5016E OE RouteD exiting abnormally, routing application already active.

Explanation: OROUTED (OE RouteD) terminated because a routing application is already active and only one routing application can be active at any time.

System Action: OROUTED (OE RouteD) initialization is terminated.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the correct routing application is active. And restart OROUTED (OE RouteD).

Source Data Set: EZRDMAI

Procedure Name: main

EZZ5019I *action* **multicast group** *group* **on interface** *interface*

Explanation: OROUTED is adding/dropping membership to a multicast group on the specified interface. Membership to a multicast group allows OROUTED to receive those multicast datagrams over that interface. The RIP Version 2 multicast address is 224.0.0.9. RIP Version 1 does not use multicast.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDIF

Procedure Name: if_mcAddMembership, if_mcDropMembership

EZZ5020E User not RACF authorized to start OE RouteD

Explanation: The user who attempted to start the OROUTED (OE RouteD) application is not RACF authorized to start it. The user must have RACF authority to the entity MVS.ROUTEMGR.OROUTED.

System Action: OROUTED (OE RouteD) initialization is terminated.

User or Operator Response: None.

System Programmer Response: Provide RACF authority to the user described in the *z/OS Communications Server: IP Configuration Guide*.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ5021I Supply suppressed for interface because a route filter has suppressed route advertisements on this interface.

Explanation: OROUTED received a request for tables on this interface but supply was disabled due to a filter in GATEWAYS data set or file. No route advertisement will take place on this interface.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS file if this is not the desired result of filtering and restart OROUTED

Source Data Set: EZBRDINP

Procedure Name: rip_input

EZZ5022I toall: Interface *interface* skipped - not broadcast or multicast capable

Explanation: The interface *interface* is skipped because there is no broadcast address or destination address to which to send requests or updates. This is a normal condition for interfaces that do not support link level broadcasting or multicasting.

System Action: OROUTED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDOUT

Procedure Name: toall

EZZ5023I OROUTED has created an OMPROUTE profile using current configuration files. Manual configuration tasks might remain.

Explanation: For customers migrating from OROUTED to OMPROUTE, OROUTED was started or modified with the **-c** parameter. Using OROUTED configuration files, OROUTED created a file to be used as a profile when starting OMPROUTE.

System Action: The OMPROUTE profile is created.

User or Operator Response: Please read the new OMPROUTE profile to determine if manual taks remain. Start OMPROUTE with the new profile.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ5024E OMPROUTE profile creation failed - File *filename* - *description* - [*errnojr*]

Explanation: OROUTED was started or modified with the **-c** parameter and a filename that could not be opened. The open of the file name failed for the reason shown in the *errnojr*.

filename is the name of the conversion output file.

description describes the meaning of the Return Code.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4–byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: An OMPROUTE profile is not created.

User or Operator Response: A unique filename must be supplied each time that OROUTED is run with **-c** or the old file must be erased each time. If no filename is specified, CNVROUTED.PROFILE will be used. Rename the current OMPROUTE profile that was created by OROUTED or start OROUTED again with **-c** and a unique filename.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main, do_modify

EZZ5025I When the -c start parameter is specified, OROUTED will not modify the IP route table and will not send or receive RIP messages nor process RIP information.

Explanation: The -c start parameter directs OROUTED to convert it's configuration files into an OMPROUTE configuration file. Once the configuration conversion is complete, OROUTED will terminate. Therefore, OROUTED will not process any RIP information and will not update the IP stack routing table.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBRDMAI

Procedure Name: main

EZZ5026I Socket option cannot be enabled

Explanation: An attempt was made to set a socket option to include local interface information when receiving data. This message will be followed by message EZZ4983I that contains the ERRNO and ERRNO2 error code information.

System Action: OROUTED ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center with ERRNO and ERRNO2 information from message EZZ4983I.

Source Data Set: EZBRDMAI

Procedure Name: get_socket

EZZ5027E OMPROUTE profile creation failed - File *filename* exists

Explanation: OROUTED was started or modified with the -c parameter and a filename that already exists.

filename is the name of the conversion output file. If no filename is specified, CNVROUTED.PROFILE will be used.

System Action: An OMPROUTE profile is not created.

User or Operator Response: A unique filename must be supplied each time that OROUTED is run with -c. Rename or erase the current OMPROUTE profile that was created by OROUTED or start OROUTED again with -c and a unique filename.

System Programmer Response: None.

Source Data Set: main, do_modify

Procedure Name: EZBRDMAI

Chapter 6. EZZ6*xxxx* messages

EZZ6002I TELNET STORAGE FAILURE FOR - *text*

Explanation: The Telnet Server attempted to allocate the indicated required storage but none is available.

text is one of the following:

MASTER CONTROL BLOCK, RSN =*n*
The Telnet Server attempted to obtain storage for its main control block but none is available.

CLIENT CONNECTION BLOCK, RSN =*n*
The Telnet Server attempted to obtain storage for a Client Connection Block but none is available.

CONN DROP PROCESS, RSN = *n*
The Telnet Server attempted to drop a connection but could not obtain the storage necessary to complete the drop process. Stopping the port might clean up the connections. Use the VARY TCPIP,,OBEYFILE command to restart the port.

TCPIP TOKEN SERVICES, RSN =*n*
The Telnet Server attempted to obtain storage for the client connection token services but none is available.

System Action: Telnet Server does not start or the connection request fails.

User or Operator Response: Contact the system programmer to diagnose this error.

System Programmer Response: If this error continues to occur after confirming that there is enough storage in your system, contact your IBM Software Support Center.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6003I TELNET *text*

Explanation: The Telnet Server is currently in the indicated state for this port.

text is one of the following:

LISTENING ON PORT *n*
Telnet Server initialization is complete for the indicated port and is listening on that port.

QUIESCED ON PORT *n*
Telnet Server quiesced for the indicated port.

Existing connections continue to support traffic. To resume new connection acceptance, issue
`VARY TCPIP,,TELNET,RESUME,PORT=n`

RESUMED ON PORT *n*
Telnet Server resumed for the indicated port.

System Action: If the port was quiesced automatically because of an accept failure, the server will automatically attempt to resume the port periodically.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6005I TELNET COMMAND TASK *text*

Explanation: The Telnet Configuration task failed to set up correctly. See specific reason for details.

text is one of the following:

© Copyright IBM Corp. 1994, 2002

ATTACH FAILED, RSN =*n*
The Telnet Server attempted to create its command control task but the MVS ATTACH service failed.

FAILED TO INITIALIZE, RSN =*n*
The Telnet Server Command Control task failed to initialize.

RESTART ATTEMPTED, RSN =*n*
The Telnet Server Command Control task failed and is attempting to restart.

TERMINATED, RSN =*n*
The Telnet Server Command Control task terminated unexpectedly during Telnet initialization.

System Action: Telnet Server profile and command processing might not be available.

User or Operator Response: Contact the system programmer to diagnose this error.

System Programmer Response: If unable to determine the cause of the task failure, contact the IBM Software Support Center.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6006I TELNET CANNOT LISTEN ON PORT *n*, *text*

Explanation: The Telnet Server attempted to listen on a port but is unable to do so.

text is the following:

CONNECTION MANAGER TERMINATED, RSN =*n*
The Telnet Server port listening task terminated unexpectedly.

System Action: Telnet Server command processor is active but will not accept client connections on the indicated port.

User or Operator Response: Contact the system programmer to diagnose this error.

System Programmer Response: If unable to determine the cause of the task failure, contact your IBM Software Support Center.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6007I TELNET LU/PRT GROUP lugroup REACHED pct % OF CAPACITY

Explanation: The number of LUs in use from this LU group reached the specified capacity warning level. The limit is specified as a percentage of the total number of LUs in the group. Once this threshold is reached, the message will not display again until the in-use count drops 10% of the total below the threshold amount. For example, a group of 200 LUs with a capacity warning level of 75% will report meeting the threshold when 150 LUs are in use. When the number of in-use LUs drops below 130 LUs, Telnet will report again when the in-use count reaches 150. If the in-use count drops to only 140 and then rises over 150, no message will be issued. This is done to reduce the messages issued when the in-use count moves slightly below and above the threshold amount.

System Action: When the LuGroup pool becomes full, connections using this LU group will be rejected.

User or Operator Response: Contact the System Programmer.

System Programmer Response: Add LUs to the LuGroup and update Telnet by issuing the VARY TCPIP,,OBEYFILE command.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6008I TELNET STOPPING

Explanation: The Telnet Server was requested to stop because TCP/IP is stopped.

System Action: Telnet Server ends. TCP/IP continues to shut down.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6009I TELNET SERVER STOPPED

Explanation: The Telnet Server terminated.

System Action: Telnet Server terminated.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6010I TELNET SERVER ENDED FOR PORT *n*

Explanation: The Telnet Connection manager task ended for the port indicated. Telnet is no longer accepting connection requests for this port.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6011I TELNET *requested_service* **FAILED, RC =** *rc* **RSN =** *rsn*

Explanation: z/OS UNIX System Services callable service was unable to complete successfully.

requested_service is the name of the requested service to perform.

rc is the return code of the requested service.

rsn is the reason code of the requested service.

System Action: None.

User or Operator Response: Contact the system programmer to diagnose the problem.

System Programmer Response: Refer to the *z/OS UNIX System Services Messages and Codes* for an explanation of the *rc* and the *rsn*.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6012I TELNET *reqserv reqfunc* **FAILED, RC =** *n* **RSN =** *n*

Explanation: z/OS UNIX System Services callable service was unable to complete successfully.

reqserv is the name of the requested service to perform.

reqfunc is the name of the requested function to perform.

rc is the return code of the requested service.

rsn is the reason code of the requested service.

System Action: None.

User or Operator Response: Contact the system programmer to diagnose the problem.

System Programmer Response: Refer to the *z/OS UNIX System Services Messages and Codes* for an explanation of the *rc* and the *rsn*.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6015I TELNET CONFIGURATION CONNECTION LOST

Explanation: Telnet lost the connection to the TCP/IP configuration component.

System Action: Periodically, the Telnet Server will attempt to regain the connection automatically.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center if the problem persists.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6017I TELNET ALREADY *requested_service* **ON PORT** *n*

Explanation: Telnet is already in the desired state, no action taken.

requested_service is the name of the requested service to perform.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6018I TELNET PROFILE *text*

Explanation: The Telnet Server profile was read in and updated successfully.

text is one of the following:

UPDATE COMPLETE FOR PORT *n*
The Telnet Profile was updated.

TESTMODE COMPLETE FOR PORT *n*
The Telnet Profile TESTMODE is complete.

System Action: Telnet Server and TCP/IP continue to run.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6022I TELNET CONFIGURATION RECONNECT *text*

Explanation: Telnet lost its connection with the TCP/IP Configuration task and is attempting to reconnect.

text is one of the following:

SUCCESSFUL
The Telnet Server was able to reestablish a connection with TCP/IP Configuration. Telnet profiles and commands can again be processed.

IN PROGRESS
The Telnet Server is attempting to reestablish a connection with TCP/IP. A connection is required to process Telnet profiles or commands. If reconnection does not complete, contact the IBM Software Support Center.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6023I TELNET CODEPAGE CONVERSION FROM *ascii_codepage_name* **TO** *ebcdic_codpage_name* **FAILED:** *text*

Explanation: The requested ASCII to EBCDIC codepage conversions failed.

ascii_codepage_name is either the first name coded on the CODEPAGE statement or the default.

ebcdic_codpage_name is either the second name coded on the CODEPAGE statement or the default.

text is one of the following:

START ERROR
The Telnet Server encountered an error starting codepage translation.

TRANSLATION ERROR
The Telnet Server encountered an error performing codepage translation.

System Action: None.

User or Operator Response: Contact the system programmer to diagnose this error.

System Programmer Response: Verify that the codepages specified are correct, compatible, and on your system.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6024I TELNET CODEPAGE CONVERSION FROM *ascii_codepage_name* **TO** *ebcdic_codpage_name* **WAS SUCCESSFUL.**

Explanation: Telnet Server successfully translated the Codepages.

ascii_codepage_name is either the first name coded on the CODEPAGE statement or the default.

ebcdic_codpage_name is either the second name coded on the CODEPAGE statement or the default.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6026I TELNET TRANSFORM INITIALIZED ON PORT *n*

Explanation: The Telnet Server DBCS Transform was successfully initialized.

System Action: Telnet Server and TCPIP continue to run.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6027I TELNET TRANSFORM INITIALIZATION FAILED, RC: *rc*

Explanation: The Telnet Server DBCS Transform failed initialization. The most probable reasons are:

- The transform load module could not be found or loaded.
- The error message file, TNDBCSER DD statement, is missing.

System Action: Telnet Server and TCPIP continue to run.

User or Operator Response: Contact the system programmer to diagnose this error.

System Programmer Response: Verify that the transform load module or the error message file can be found by Telnet.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6028I TELNET TRANSFORM HAS ENDED

Explanation: The Telnet Server DBCS Transform ended either normally or abnormally.

System Action: Telnet Server and TCPIP might not continue to run.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6034I TELNET CONN *connid* **LU** *luname action object*

Message Format:

```
EZZ6034I TELNET CONN connid LU luname action object
         IPADDR..PORT ipaddr            port module
```

Explanation: This message gives a summary status for all connections.

connid is the TCP/IP connection ID.

luname is the name of the LU representing the client. To avoid flooding your console, the DEBUG option EXCEPTION will set the *luname* to **MULTIPLE** if *action* is **CONN DROP** and more than one connection was dropped for the same reason within a 15 second interval.

action is one of the following:

ACCEPTED
A connection request was accepted on the server port indicated.

NEGOTIATED
Connection negotiation finished and the connection type is specified.

IN SESSION
A session was established on this connection. The Appl name is specified.

SESS DROP
The session on this connection was dropped for one of the reasons listed under object.

CONN DROP
The connection was dropped for one of the reasons listed under object. The **CONN DROP** message is issued for error conditions and inactivity reasons whether or not DEBUG is coded. If DEBUG EXCEPTION or NO DEBUG is coded, and more than one connection is dropped for the same reason within 15 seconds, a single message with *luname* of MULTIPLE will be issued. This is done to reduce console flooding. If you want to

see a message for each connection being dropped, specify the **DEBUG SUMMARY** statement. Refer to the *z/OS Communications Server: IP Configuration Guide* for more information about Telnet Diagnostics.

object is one of the following:

- If *action* is **ACCEPTED**, *object* is the Telnet server port number.
- If *action* is **NEGOTIATED**, *object* is the connection mode.
- If *action* is **IN SESSION**, *object* is the name of the host application.
- If *action* is **SESS DROP** or **CONN DROP**, *object* is one of the following reasons:

ABEND An ABnormal END occurred in the Telnet code.

CLNTDISC The end user or client emulator disconnected from the connection.

CLOSEERR An error occurred during close processing.

ERR *code* A telnet error occurred. See EZZ6035I for a complete list of the return codes.

INACT-K The KEEPINACTIVE timer detected no session activity for the specified time.

INACT-P The PRTINACTIVE timer detected no session activity for the specified time.

INACT-S The INACTIVE timer detected no session activity for the specified time.

LUXABEND An ABnormal END occurred in LUEXIT code used by the connection.

NSEXIT The Telnet LU NSEXIT is being driven because of session breakage.

STOPPORT The port was stopped by an operator command.

STOPTRAN The Transform task was ended by an operator command or an error.

SYSLGOFF The end user issued SYSREQ LOGOFF and LUSESSIONPEND was not specified.

TIMEMARK A TIMEMARK was not answered by the client in the specified time indicating a lost connection.

TKOVER The connection and session are being ended because it is being taken over by another connection.

TKOVER-R The connection is being ended because it is being taken over by another connection. The session remains active.

TPEND The Telnet LU TPEND exit is being driven because of a forced close of the host application.

TRANCLOS The Transform task found an error and is closing the connection.

UNBIND The end user issued a normal LOGOFF from the host application.

UNBIND02 The session ended but issued a CLSDST-PASS to another application.

UNBINDQS The session ended but it is in a chain of QSESSION sessions. Telnet will wait for one of the earlier sessions to initiate another session.

UNKNOWN Close is occurring for some reason other than any listed above.

USSLGOFF The end user issued LOGOFF from the USS screen.

ipaddr..port is the client IP address and port number.

module is the detecting module if the connection was dropped because of a Telnet error.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6035I TELNET DEBUG *ipaddr..port*

Message Format: This message can appear in the following formats:

- When a Detail Debug message is issued for a connection related problem:

```
EZZ6035I TELNET DEBUG CLIENT IPADDR..PORT ipaddr..port
 CONN: connid LU: luname MOD: modname
 RCODE: rcode-instance description
 PARM1: parm PARM2: parm PARM3: parm
```

- When a Trace Debug message is issued for the one connection being traced:

```
EZZ6035I TELNET DEBUG CLIENT IPADDR..PORT ipaddr..port
 CONN: connid LU: luname MOD: dir
  dir tracedata
 PARM1: parm PARM2: parm PARM3: parm
```

- When a Profile error is found and either a warning or error message is issued:

```
EZZ6035I TELNET DEBUG PROFILE msg_type,LINE:line MOD: modname
 RCODE: rcode-instance description
 PARM1: parm PARM2: parm PARM3: parm
```

Explanation: A diagnostic message was displayed for debugging purposes. If ″Contact the IBM Software Support Center″ is recommended, a dump of TCP/IP and its dataspaces will be needed. If the problem is re-creatable, a CTRACE with option TELNET only is very helpful.

ipaddr..port is the client IP address and port number if appropriate.

msg_type is either of the following:

Error The profile statement is in error and the affected port will not be started or updated.

Warning
The profile statement is in error and will be ignored. The affected port update will continue without this statement

connid is the connection ID assigned by the TCPIP stack.

luname is the name of the Telnet LU representing the client.

dir is the direction of the data flow.

tracedata is the first 48 bytes of data sent or received from the client or the VTAM application. The RPL is included, if applicable.

modname is the name of the module reporting the error. For trace entries, this field is used as a source/destination field.

rcode is the return code. The code might indicate an error or it might be normal processing. See the descriptions below for details.

instance is the instance number of the error in the module.

parm is an extra parameter to pass additional information based on the return code.

description is one of the following:

0000 OK
No errors encountered.

0008 Storage obtain request failed.
Causes include low on storage condition or passing invalid parameters on the storage request. Verify storage availability. In some cases, Parm1 contains the size of the storage request. If storage is available contact the IBM Software Support Center.

0009 Storage release request failed.
Causes include trying to free the same storage more than once or passing invalid parameters on the storage request. If a storage release failure occurs, contact the IBM Software Support Center.

000A IOCTL request failed.
Telnet issues an IOCTL request to update information used by the NETSTAT display command. Parm1, Parm2, and Parm3 might be set to the IOCTL return value, return code, and reason code respectively. If an IOCTL failure occurs, contact the IBM Software Support Center.

000B Available return code.
This return code is not currently used and is available for future use.

000C Timer request failed.
Telnet requested a timer and the request failed. Contact the IBM Software Support Center.

000D Lock obtain request failed.
A lock obtain failure is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

000E Lock release request failed.
A lock release failure is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

000F CVB is invalid.
The CVB control block represents the client connection. An invalid CVB is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

0010 Allocation of a message control block failed.
Message control blocks are used to move data traffic between VTAM and the client. If an allocation failure occurs, contact the IBM Software Support Center.

0011 Work element Queue and Post request failed.
Internal Telnet work element processing failed. Contact the IBM Software Support Center. PARM1 might contain an additional failure code to assist IBM Software Support to resolve the problem.

0012 Internal list request failed.
Internal list processing failed. Contact the IBM Software Support Center.

0013 Available return code.
This return code is not currently used and is available for future use.

0014 CVB lock obtain request failed.
Lock processing of the CVB control block failed. This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

0015 CVB lock release request failed.
Lock processing of the CVB control block failed. This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

0016 CVB token is invalid.
The token for lock processing of the CVB control block is invalid. This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

0017 Available return code.
This return code is not currently used and is available for future use.

0018 Available return code.
This return code is not currently used and is available for future use.

0019 CVB token does not match master token.
The CVB token used by a particular process does not match the token stored in the Telnet tables. This is sometimes part of normal processing when a session or connection is being dropped. If reported at other times, contact the IBM Software Support Center.

001A Telnet timer has been cancelled.
The timers used for INACTIVE and SCANINTERVAL are cancelled when the port is being stopped. If reported at other times, contact the IBM Software Support Center.

001B Takeover target is invalid.
Takeover was attempted on a target that cannot be taken over. A probable cause is that the profile used by the target did not specify takeover. Issue a detailed display of the original target connection. Check the profile flags to see if takeover is supported. If takeover is supported, contact the IBM Software Support Center.

001C Copy of a message control block failed.
Message control blocks are used to move data traffic between VTAM and the client. If a copy failure occurs, contact the IBM Software Support Center.

001D Duplication of a message control block failed.
Message control blocks are used to move data traffic between VTAM and the client. If a duplication failure occurs, contact the IBM Software Support Center.

001E Internal tree process failed.
Internal Patricia tree processing failed during registration work. A possible cause is a takeover timing condition. Retry the connection. If the problem persists, contact the IBM Software Support Center. PARM1 might contain an additional failure code to assist IBM Software Support to resolve the problem.

001F OE Dub Process failed.
Telnet initialization for socket setup failed with the BPX Dub process. Contact the IBM Software Support Center.

0020 Dynamic LU tree creation failed.
The creation of the dynamic LU tree needed to track LU usage failed during Telnet initialization. Contact the IBM Software Support Center.

0021 Patricia Tree ADD failed for IP node.
Internal Patricia Tree processing failed during profile IP node processing. An internal return code is saved in Parm1. If the problem persists, contact the IBM Software Support Center.

0022 Patricia Tree DELETE failed for IP node.
Internal Patricia Tree processing failed during profile IP node processing. An internal return code is saved in Parm1. If the problem persists, contact the IBM Software Support Center.

0023 Patricia Tree CREATE failed for IP node.
Internal Patricia Tree processing failed during profile IP node processing. An internal return code is saved in Parm1. If the problem persists, contact the IBM Software Support Center.

0024 Patricia Tree token is invalid.
Internal Patricia Tree processing failed during profile IP node processing. An internal return code is saved in Parm1. If the problem persists, contact the IBM Software Support Center.

0025 Takeover target is closing.
The target for connection or session takeover is in the processing of closing. The takeover will not occur. The taker should retry the connection request after receiving this message.

0026 Load of EZBTTMST load module failed.
Telnet Connection Manager load module could not be loaded. The most probable reason is that the load module is not part of the system library that is accessible to Telnet. Parm1 might contain the system completion code and Parm2 might contain the reason code. If the load module is accessible to Telnet, contact the IBM Support Center.

0027 Load of EZBTPGUE load module failed.
Telnet User Exit Interface load module could not be loaded. The most probable reason is that the load module is not part of the system library that is accessible to Telnet. Parm1 might contain the system completion code and Parm2 might contain the reason code. If the load module is accessible to Telnet, contact the IBMSupport Center.

1001 Client disconnected from the connection.
The end user or client emulator initiated ending the connection by disconnecting. If this return code is unexpected, analyze the client to determine why the client initiated a disconnect.

1002 Close socket request failed.
This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

1003 A TCP/IP receive data request failed.
This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

1004 A close request is already in progress.
A second close was attempted. The first close will continue and the second close will be ignored. If the first close does not complete, contact the IBM Software Support Center.

1005 A Cancel socket I/O request failed.
This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

1006 A TCP/IP send data request failed.
This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

1007 Socket fastpath setup failed.
Telnet connection sockets are defined as fastpath to improve performance. If fastpath setup failure occurs, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

1008 A Get Hostname by IP address request failed.
Telnet attempted to find the hostname of the client but failed. The most probable cause is incorrect setup of the system DNS. Ensure that the correct TCPIP.DATA statements are being used. Telnet uses the MVS search order. See the *z/OS Communications Server: IP Configuration Guide* for information about the TCPIP.DATA file and search order. If this return code is unexpected, contact the IBM Software Support Center.

1009 Available return code.
This return code is not currently used and is available for future use.

100A Connection type of NONE was specified.
The profile option CONNTYPE NONE was specified indicating no connections are allowed. If this is not the intended result, reconfigure the Telnet profile.

100B Unexpected SSL handshake encountered.
An SSL handshake header was encountered on a basic port or the client immediately entered an SSL handshake for a CONNTYPE other than SECURE or ANY. Verify that the client and port settings are compatible.

100C A TCP/IP send immediate request failed.
This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

100D TCP/IP async send did not complete immediately.
A probable cause is a blocked socket. This condition should affect only the client that cannot accept additional data. If the entire server is affected, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

100E The transform task is not available.
The DBCS transform task is not available to perform the requested transformations. Either the DBCSTRANSFORM statement is missing, the load module did not get loaded, or the load module was removed because of an error. If an error occurred, contact the IBM Software Support Center.

100F A send was issued without any data.
A request to send data to the client was issued but there was no data specified. Contact the IBM Software Support Center.

1010 The socket was dropped.
This is most probably done by the operator. If not, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

1011 The transform request in Parm1 is invalid.
The transform request is not a valid request that can be handled by the transform process. Contact the IBM Software Support Center. The request is reported in Parm1.

1012 A client doing a takeover is closing.
A new client begun the close process while waiting for the takeover timer to expire. The new connection will finish closing after the takeover timer expires. If this return code is unexpected, contact the IBM Software Support Center.

1013 Takeover already in progress.
A new client attempted to take over a connection that already is being taken over by another connection. Only one connection at a time can try to take over a connection.

1014 Takeover has started.
A new client began the takeover process. This return code is for tracking purposes and does not indicate a problem. Parm1 contains the takeover time value specified on the TKOSPECLU or TKOSPECLURECON parameter in hexadecimal form.

1015 Takeover not specified on original client PROF.
A new client attempted to take over a connection but the original connection does not allow takeover. Takeover must be specified on the profile used by the original connection. The new client might not be attempting takeover and instead accidentally chose an LU already in use. In this case, the client should use a different LU name.

1016 Port Task setup failed.
The setup of the Port task failed. The port will not be available. The task was set up, Port task code began to run, but an error occurred in initialization. Parm1 contains the Port task return code. Parm2 contains the hex value of the port number. Contact the IBM Software Support Center.

1017 Attach of the Port task failed.
The MVS macro, ATTACH, failed to attach the Port task during Telnet initialization. Parm1 contains the ATTACH return code. PARM2 contains the hex value of the port number. Contact the IBM Software Support Center.

1018 The Port task has ended in error.
The Port task ended because of an error condition. Instance 01 indicates that the task was set up properly and at some later time, an error occurred. Parm1 contains the Port task return code. Instance 02 indicates that the task was set up, but the Port task code never ran. Parm1 contains a system completion code. Instance 03 indicates that the task was set up, Port task code was initialized, but an error quickly occurred. Parm1 contains the Port task return code. In all cases, Parm2 contains the hex value of the port number. Contact the IBM Software Support Center.

1019 The connection ID could not be obtained.
The request by Telnet to get the connection ID for this connection failed. The connection request will be failed. Contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

101A Setup of Out Of Band data handling failed.
The request by Telnet to handle Out Of Band data inline failed. The connection request will be failed. Contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return value, return code, and reason code respectively, which are defined in *z/OS UNIX System Services Messages and Codes*.

101B The linkname table could not be obtained.
The request by Telnet to get the TCP/IP stack linkname table failed. The connection will continue but any profile mappings based on linkname will fail. Contact the IBM Software Support Center.

101C Send data to the client.
The Debug Trace option was selected causing trace messages for one connection. Parm1 contains the total length sent.

101D Receive data from the client.
The Debug Trace option was selected causing trace messages for one connection. Parm1 contains the total length received.

101E The profile control block could not be found.
The most probable cause is that all the port profiles are qualified and the connection request has a different destination IP address or linkname than any that are defined. This might be working as designed if only the qualified port destinations are to be used. Another solution would be to create a non-qualified profile to cover unexpected connections.

101F There are no current profiles available.
There are no profiles available for the connection to use. This error should not occur. If it does, contact the IBM Support Center.

1020 The main connection CB could not be obtained.
The control block structure that manages connections could not be obtained. The most probable reason is that storage is not available. Verify that storage is available. If a storage shortage is not the problem, contact the IBM Support Center.

1021 The takeover connection is now invalid.
During the closing of the connection being taken over, the takeover connection became invalid. The most probable reason is that the connection is being disconnected by the end user. PARM1 might contain an additional failure code to assist IBM Software Support to resolve the problem.

1022 Available return code.
This return code is not currently used and is available for future use.

1023 Telnet does not support the exit type.
The Telnet common exit processor was given control to process an exit type other than an Interpret exit or LU name exit. This should not occur. Contact the IBM Support Center. PARM1 contains the exit type code that was attempted.

1024 Adding Userid information for keep LU failed.
During LU unassignment, the Client Identifier User ID information could not be saved. The most probable reason is a storage shortage. If a storage shortage is not a problem, contact the IBM Support Center.

1025 Adding hostname information for keep LU failed.
During LU unassignment, the Client Identifier hostname information could not be saved. The most probable reason is a storage shortage. If a storage shortage is not a problem, contact the IBM Support Center.

1026 Adding IP addr information for keep LU failed.
During LU unassignment, the Client Identifier IP address information could not be saved. The most probable reason is a storage shortage. If a storage shortage is not a problem, contact the IBM Support Center.

1027 The Stack aborted the connection due to no ACK.
The request by Telnet to send to the client did not get an ACK in the maximum retry limit. The connection is reset by the Stack.

2001 ACB mismatch during Bind processing.
The ACB address in the bind does not match the ACB representing the connection. If the application does not appear to be at fault, contact the IBM Software Support Center. A VTAM internal trace in addition to the Telnet debug information will be needed.

2002 Available return code.
This return code is not currently used and is available for future use.

2003 Available return code.
This return code is not currently used and is available for future use.

2004 Available return code.
This return code is not currently used and is available for future use.

2005 The session is not a SNA session.
Session verification indicates this is not a SNA session but is attempting to perform a SNA-type function. If SNA function is desired, change the Devicetype logmode.

2006 Error writing SMF record.
An error occurred attempting to write an SMF record. Contact the IBM Software Support Center. PARM1 might contain the SMF return code to assist IBM Software Support to resolve the problem.

2007 VTAM macro RECEIVE failed.
This is sometimes part of normal processing when a session or connection is being ended. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2008 A response to VTAM failed.
An error occurred while attempting to send a response to VTAM. This is sometimes part of normal processing when a session or connection is being ended. If reported at other times, contact the IBM Software Support Center.

2009 Sending UNBIND to client failed.
A probable cause is a BIND was not already sent to the client. Verify that the application is sending the correct sequence of BINDs or UNBINDs or both. If the application does not appear to be at fault, contact the IBM Software Support Center. PARM1 might contain an additional return code indicating why the SEND failed.

200A NSEXIT was driven for a BIND failure.
The Telnet LU Network Services exit was driven indicating a BIND request failed. Review VTAM messages for possible causes.

200B Sending BIND to client failed.
A probable cause is a BIND was already sent to the client. Verify that the application is sending the correct sequence of BINDs or UNBINDs or both. If the application does not appear to be at fault, contact the IBM Software Support Center. PARM1 might contain an additional return code indicating why the SEND failed.

200C RPLRQR,RPLSTSN not valid for TS profile 2/3.
RPLRQR and RPLSTSN are not valid for either TS profiles 2 or 3. Since these are the only profiles that the Telnet server supports this case should not occur. If it does, analyze why the host application is sending these requests and change the application.

200D Received BIND while already bound.
Verify that the application is sending the correct sequence of BINDs/UNBINDs. If the application does not appear to be at fault, contact the IBM Software Support Center.

200E Invalid TERMSESS type encountered.
The VTAM macro TERMSESS was requested with an invalid internal function code. Contact the IBM Software Support Center.

200F VTAM macro TERMSESS failed.
This is sometimes part of normal processing when a session or connection is being ended. If reported at other times, contact the IBM Software Support Center. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2010 VTAM macro OPNSEC failed.
Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2011 VTAM macro REQSESS failed.
Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2012 VTAM macro CLOSE ACB failed.
Parm1, and Parm2 might be set to return code and reason code respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2013 VTAM macro OPEN ACB failed.
Parm1, Parm2, and Parm3 might be set to return code, reason code, and ACB error flag value respectively, which are defined in *z/OS Communications Server: SNA Programming*. The most probable reason is that the LU is not active in VTAM. Issue D NET,ID=luname to see the VTAM status of the LU.

2014 VTAM macro SETLOGON failed.
Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk,2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2015 NSEXIT was driven for session cleanup.
A probable cause is that the host application was inactivated. Review VTAM messages for other possible causes.

2016 NSEXIT was driven for a CINIT failure.
The Telnet LU Network Services exit was driven indicating a CINIT request failed. Review VTAM messages for possible causes.

2017 NSEXIT was driven for a CTERM failure.
The Telnet LU Network Services exit was driven indicating a CTERM request failed. Review VTAM messages for possible causes.

2018 NSEXIT was driven for an UNBIND failure.
The Telnet LU Network Services exit was driven indicating an UNBIND request failed. Review VTAM messages for possible causes.

2019 Session does not exist.
Session verification determined that a session no longer exists. The attempted function is not done. This is sometimes part of normal processing when a session is ended. If reported at other times, contact the IBM Software Support Center.

201A Session data queue is being purged.
The host application sent a CLEAR to purge the data queue. New data cannot be added until the CLEAR is complete.

201B Available return code.
This return code is not currently used and is available for future use.

201C Available return code.
This return code is not currently used and is available for future use.

201D Available return code.
This return code is not currently used and is available for future use.

201E Available return code.
This return code is not currently used and is available for future use.

201F Available return code.
This return code is not currently used and is available for future use.

2020 VTAM RECEIVE macro requested invalid function.
RPL verification determined that an unrecognized function was attempted during the VTAM RECEIVE process. Verify that the application is sending valid RPL requests. If the application does not appear to be at fault, contact the IBM Software Support Center. A VTAM internal trace in addition to the Telnet debug information will be needed.

2021 Available return code.
This return code is not currently used and is available for future use.

2022 Available return code.
This return code is not currently used and is available for future use.

2023 Retry scheduled for RPL request.
A probable cause for the failure is a temporary storage shortage in VTAM. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2024 RPL length specified but RPL area is zero.
RPL verification determined that the RPL length field is set but the RPL area field is zero. Storage corruption is the probable reason. Contact the IBM Software Support Center.

2025 Available return code.
This return code is not currently used and is available for future use.

2026 Available return code.
This return code is not currently used and is available for future use.

2027 Maximum retries exceeded for VTAM RPL.
A probable cause for the failure is a temporary storage shortage in VTAM. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2028 Available return code.
This return code is not currently used and is available for future use.

2029 VTAM RPL posted with non-zero RPLrtncd/RPLfdbk2.
A minor error was reported in a VTAM RPL request. The session will not be ended. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

202A VTAM RPL returned negative response.
A negative response was returned from a VTAM RPL macro. The session will be ended. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

202B Available return code.
This return code is not currently used and is available for future use.

202C Available return code.
This return code is not currently used and is available for future use.

202D VTAM macro SEND expedited failed.
SHUTC and SIGNAL are expedited RPLs. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

202E VTAM macro REQSESS failed. Already in session.
Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

202F BIND for printer received with invalid LU.
The BIND received from the host application did not specify an LU1 or an LU3. Determine why the application sent the invalid BIND.

2030 Sending a response to VTAM when none expected.
The client sent a response to Telnet that should be passed through to VTAM. In this case, Telnet does not think the application should be sent a response. This is sometimes part of normal processing when a connection is being dropped. If reported at other times, contact the IBM Software Support Center.

2031 Abnormal termination of request.
VTAM abnormally terminated a request because of an error detected while processing the request or because of a session, task, or address space error. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2032 Receive negative response and continue process.
A negative response was received from the host application with a sense code indicating the session should not be ended.

2033 Send data attempted without having SDT.
A send data request is rejected because a Start Data Traffic (SDT) was not received. Either the initial SDT was not received or a CLEAR was received and a new SDT was not received.

2034 Specified maximum ReqSess attempts exceeded.
The connection appears to be in a CLSDST PASS loop. The number of Request Session attempts in a 10 second period exceeded the number specified on the MAXREQSESS statement or the default. Parm1 contains the limit value in hexadecimal. The count is incremented when a BIND is received from the host application.

2035 UNBIND or CLEAR ended a RECEIVE request.
The VTAM Application issued an UNBIND or CLEAR request that ended the RECEIVE RPL request. The connection is kept, waiting for the follow-up process from the application. Parm1, Parm2, and Parm3 might be set to return code, reason code, and RPLrtncd/RPLfdbk2 respectively, which are defined in *z/OS Communications Server: SNA Programming*.

2036 Send data to the VTAM application.
The Debug Trace option was selected causing trace messages for one connection. Parm1 contains the total length sent. Parm2 contains the RH fields.

2037 Receive data from the VTAM application.
The Debug Trace option was selected causing trace messages for one connection. Parm1 contains the total length sent. Parm2 contains the RH fields.

2038 BIND specifying delayed response mode received.
The BIND received from the host application specifies delayed response mode. Although this BIND will be accepted, the Telnet server does not support delayed response mode. Multiple outstanding requests for definite responses from the host application can lead to session hangs. Determine why the application sent the BIND with delayed response mode specified.

3001 The LU is in use and can not be inactivated.
The LU name being inactivated is in use by a Telnet connection. The LU name cannot be inactivated while in use.

3002 The LU is already inactive.
The LU name being used is inactive.

3003 LUs are all in use.
The specific LU requested is in use or a generic request mapped to an LUGROUP or DEFAULTLUS pool that has no LUs available. Verify that the profile mapping statements are correct and that sufficient LUs are available. TN3270E connections are assigned LUs whether or not a session is established. Be sure to account for this additional LU usage by adding additional LUs if necessary.

3004 LU is not available.
There is no LU mapping for this client or, if a specific connection request, there is no LU definition that matches the LU name on the request. Verify that the profile mapping statements are correct.

3005 Database error - Invalid database header.
An invalid database header was detected. A probable cause is storage corruption. Contact the IBM Software Support Center.

3006 Specific requests are only valid in TN3270E.
A specific LU connection request was attempted on a non-TN3270E connection. Verify NOTN3270E is not coded and check the client to confirm a TN3270E connection was requested. Some clients allow a specific LU request on a TN3270 connection that is not supported on this server.

3007 Invalid map index has been detected.
LU map index verification detected an error. Contact the IBM Software Support Center.

3008 Invalid application index has been detected.
Application LU map index verification detected an error. Contact the IBM Software Support Center.

3009 Available return code.
This return code is not currently used and is available for future use.

300A Invalid bundle index has been detected.
An invalid internal index was detected. Contact the IBM Software Support Center.

300B Telnet LU not in use.
Telnet attempted to make an LU available during close processing that is not registered as being in use. Contact the IBM Software Support Center.

300C Database headers in the TCFG are corrupted.
One or more database headers in the profile control block, TCFG, is corrupted. Contact the IBM Software Support Center.

300D The LU is not associated with this connection.
The close process is attempting to make an LU available that Telnet thinks should be associated with the connection. In fact, no LU was associated with the connection. Contact the IBM Software Support Center.

300E Invalid indices have been detected in the TCFG.
The profile control block, TCFG, has one or more invalid indices. A probable cause is storage corruption. Contact the IBM Software Support Center.

300F Linkname Lookup failed.
During profile processing, a client identifier was checked to see if it was a linkname. In this case the name on the profile mapping statement is not a linkname.

3010 Lookup request is invalid.
An internal lookup request is invalid. Contact the IBM Software Support Center.

3011 Application name is required.
An application name is required for a session to be established. This return code is part of normal processing whenever a USSMSG10 or solicitor panel is sent to the client. Another cause is that the end user did not enter an application name when prompted to do so.

3012 Application name is invalid.
The application name entered by the end user is not valid based on the ALLOWAPPL or RESTRICTAPPL statements in the profile. Be sure the end user is requesting a valid name. Also, be sure any applications that are the target of a CLSDST PASS are in the profile table. For example, a logon to TSO causes a CLSDST PASS to TSO0001. An ALLOWAPPL TSO* is needed for the second TSO application name to be valid.

3013 Application name has a syntax error.
The application name entered by the end user contains a syntax error. Application names must be 1-8 characters with the first character A-Z,@,#,$, and the second through eighth character A-Z,0-9,@,#,$.

3014 Userid is required.
The application name requested by the end user is a RESTRICTAPPL. The USERID specified is not listed on the RESTRICTAPPL statement.

3015 Userid and password are required.
The application name requested by the end user is a RESTRICTAPPL. It requires the entry of a USERID and password before the session can be established.

3016 Password is required.
The application name requested by the end user is a RESTRICTAPPL. It requires the entry of a password before the session can be established.

3017 Password is invalid.
The password entered by the end user is invalid. reenter the correct password or contact your local system or security administrator.

3018 Password is expired.
The password entered by the end user expired. The end user needs to enter a new password.

3019 Password is revoked.
The password entered by the end user was revoked. Contact your local system or security administrator.

301A Password not in the security program.
The password entered by the end user could not be found in the security system. Contact your local system or security administrator.

301B Password failed in the security program.
The password entered by the end user failed in the security system for an unknown reason. Contact your local system or security administrator.

301C Password failed in the security system group.
The password entered by the end user was not part of the security system group. Contact your local system or security administrator.

301D System security password group revoked.
The user ID entered by the end user is part of a group that was revoked. Contact your local system or security administrator.

301E Password change requires old and new password.
When changing a password, it is required that both the old and new passwords be entered.

301F New password is invalid.
When changing a password, the new password must meet certain password formatting rules. Contact your local system or security administrator for details.

3020 New password change failed.
An unknown failure occurred while trying to change the password. Contact your local system or security administrator.

3021 Associated printer already in use.
The associated printer is already in use by another telnet connection. Try specifying another printer name or wait until the other connection is dropped.

3022 Associated terminal is invalid.
The terminal LU specified on the associated printer connect request is invalid. The terminal LU name used for the association is not a valid terminal LU name. Using a client trace or DEBUG TRACE, verify that the correct terminal LU name is on the associated connect request. If it is, contact the IBM Software Support Center.

3023 Associated printer list size is incorrect.
The printer LU group and the terminal LU group must be the same size. The number of single entries must match, the number of bundle entries must match, and the number in each bundle must match. Verify that the LUGROUP and associated PRTGROUP pools do have the required one-to-one match-ups required.

3024 Available return code.
This return code is not currently used and is available for future use.

3025 System security request is invalid.
Telnet issued an invalid System Security request. Contact the IBM Software Support Center.

3026 System security STAT request failed.
CLIENTAUTH SAFCERT requested but the security product is not active. Ensure the security product is active before using CLIENTAUTH SAFCERT. Parm1, Parm2, and Parm3 contain the return code, SAF return code and SAF reason code from the RACROUTE FASTAUTH request respectively.

3027 Client not authorized to use the port.
CLIENTAUTH SAFCERT was specified for the connection. The user ID associated with the client certificate does not have read access to the port resource and the connection is closed. Parm1, Parm2, and Parm3 contain the return code, SAF return code and SAF reason code from the RACROUTE FASTAUTH request respectively. Contact your local system or security administrator if this client requires access.

3028 System security client certification failed.
CLIENTAUTH SAFCERT was specified for the connection and the client certificate is not registered with the security product. Contact your local system or security administrator if this client requires access. PARM1 contains the SAF return code.

3029 INITACEE is not available.
CLIENTAUTH SAFCERT was specified for the connection. However, the security product does not support client certificate queries. Contact your local system or security administrator.

302A Associated connect request is invalid.
The associated connect request is invalid. Probable causes include the requester is not a printer or the associated LU name is blank.

302B Associated terminal LU is inactive.
The terminal LU must active before the printer can issue an associated connect request. In this case, the LU is not active.

302C No printer group associated with Terminal LU.
An associated connect request was received that contained a terminal LU name that does not have an associated printer. A probable cause is that the terminal LU was mapped to an LUGROUP on an LUMAP statement that does not have an associated printer pool defined. Verify that the client is using an LU from an LUMAP statement that has an associated printer pool defined.

302D LU lookup confirmation failed.
Lookup is often done more than once. For example, a TN3270E connection is assigned an LU during connection negotiation. Later, another lookup is done that includes the application name. These later lookups simply confirm that the LU assigned earlier is still correct. In this case, confirmation failed. Contact the IBM Software Support Center.

302E TakeoverRecon with a different APPL attempted.
An end user is attempting a takeover with session reconnect and is specifying a different application name than the original session used. The original session is dropped and takeover is done without the reconnect function.

302F System security user profile not defined.
The user profile is not defined in the system security application. Contact your local system or security administrator.

3030 TakeoverRecon with a different USERID attempted.
An end user is attempting a takeover with session reconnect and is specifying a different USERID name than the original session used. The original session is dropped and takeover is done without the reconnect function.

3031 Specified different applname when DEFONLY coded.
The default application name on the DEFAULTAPPL, LINEMODEAPPL, or LUMAP-DEFAPPL statement has DEFONLY coded. This means the end user can only log on to that application. In this case the end user attempted to log on to a different application name from a USS screen or the solicitor panel. These screens can be sent because of a logon error, logoff of a logappl session, or logoff of a session when FIRSTONLY is coded.

3032 Invalid Client Identifier type.
An internal error caused an invalid Client Identifier to be used. Parm1 contains the hex value of the Client Identifier. Contact the IBM Support Center.

3033 Invalid Object type.
An internal error caused an invalid Object to be used. Parm1 contains the hex value of the Client Identifier. Contact the IBM Support Center.

3034 The Object group has no entries.
An Object group has no valid Object entries. PARM3 contains the group name. Determine the errors in the group and retry the Profile.

3035 The Client Identifier group has no entries.
A Client Identifier group has no valid Client Identifiers. PARM3 contains the group name. Determine the errors in the group and retry the Profile.

3036 Invalid parms were encountered.
Invalid parameters were encountered while processing the statement. Review the syntax for the statement in the *z/OS Communications Server: IP Configuration Reference*.

3037 Invalid mapping statement.
An internal error caused an invalid Mapping statement to be used. Parm1 contains the hex value of the Mapping statement. Contact the IBM Support Center.

3038 Mapping of the Client Identifier failed.
A valid Client Identifier could not be found for this mapping statement. The most probable cause is using a Group name before the group is identified.

3039 Mapping of the Object failed.
A valid Object could not be found for this mapping statement. The most probable cause is using a Group name before the group is identified.

303A The mapping statement is a duplicate.
The mapping statement is an exact duplicate of an earlier mapping statement.

303B Object mapped to Client Identifier is replaced.
DEFAULTAPPL, LINEMODEAPPL, USSTCP, and INTERPTCP allow only one Object to be mapped to a Client Identifier. An earlier statement mapped a different Object to the Client Identifier on the current line. Parm1 contains the old Object name and Parm3 contains the new Object name.

303C A hash entry was not found.
A host name, linkname, or user ID hash table entry was not found during delete processing. If the problem continues, contact the IBM Support Center.

303D Available return code.
This return code is not currently used and is available for future use.

303E Invalid LU name assigned by LU exit.
An invalid LU name was assigned by an LU exit during a connection request attempt. The invalid name appears in the LU: field of this message. The connection request being processed is terminated as a result of the invalid name. The LU exit should be investigated and corrected to avoid this error.

303F Invalid LUGROUP name.
The LU Group name specified on the ALLOWAPPL or RESTRICTAPPL statement is invalid. This is most probable because the LU Group was not defined earlier. Parm3 contains the invalid LU Group name.

3040 Multiple LUGROUPs were specified. Last one used.
Multiple LU Groups were defined on the ALLOWAPPL or RESTRICTAPPL statement. The last LU Group specified is used and is contained in Parm3.

3041 LUG parameter is used instead of single LUs.
One or more LU Groups were defined with single LUs on the ALLOWAPPL or RESTRICTAPPL statement. The last LU Group specified is used and is contained in Parm3.

3042 The LU being activated is not on inactive list.
The LU name being activated is not on the inactive list and therefore cannot be activated. Use the INACTLUS display command to see what LU names are inactive.

3043 Available for use.

3044 Available for use.

3045 Duplicate RESTRICTAPPL userid. Last one is used.
The same User ID was specified more than once on the RESTRICTAPPL statement. Parm3 contains the duplicated User ID.

3046 Available return code.
This return code is not currently used and is available for future use.

3047 Available return code.
This return code is not currently used and is available for future use.

3048 Allowappl name invalid. Already a Restrictappl.
The application name contained in Parm3 was already defined as a RESTRICTAPPL and cannot now be defined as an ALLOWAPPL. Be sure the application name is correct on each statement.

3049 Invalid Object Function.
An invalid Object function, contained in Parm1, was requested during database processing. Contact the IBM Support Center.

304A Invalid Client ID Function.
An invalid client ID function, contained in Parm1, was requested during database processing. Contact the IBM Support Center.

304B Same name Allowappl is being replaced.
An earlier ALLOWAPPL statement with the same application name contained in Parm3 is being replaced by the current statement. Be sure each statement has the correct application name specified.

304C Same name Restrictappl is being replaced.
An earlier RESTRICTAPPL statement with the same application name contained in Parm3 is being replaced by the current statement. Be sure each statement has the correct application name specified.

304D LU range lower base does not match upper base.
The LU range shown in Parm3 does not have the same base portion in the lower and upper range names.

304E LU range lower base is higher than upper base.
The lower LU name in the LU range shown in Parm3 has a higher value than the higher LU name in the range. Correct the range so that the lower LU name is lower than the upper LU name.

304F LU range variant larger than 2G. Range ignored.
The LU range shown in Parm3 will generate a range larger than 2G, which is invalid. The second character range cannot be greater than 13. Reduce the range to be less than 2G.

3050 LU range started as numeric. Can not have alpha.
The LU range shown in Parm3 is assumed to be numeric but an alphabetic character was found in the variant portion. Make the range all numeric.

3051 LU range started as alpha. Can not have numeric.
The LU range shown in Parm3 is assumed to be alphabetic but a numeric character was found in the variant portion. Make the range all alphabetic.

3052 This LUMAP replaces earlier LUMAP with same LU.
The LU or Group name contained in Parm3 is used in an earlier LUMAP statement with different parameters. The current statement replaces the earlier statement. Be sure each statement is coded correctly.

3053 Client ID already used, can not be used again.
The Client Identifier contained in Parm1 was used earlier. If used in a Group, Parm3 contains the earlier Group name. The original use is used and this entry is ignored. Be sure each was coded correctly.

3054 An internal error caused Inactivation failure.
An internal error did not allow the LU Inactivation to complete. The most likely causes are storage shortage or corruption of a data structure. Contact the IBM Support Center.

3055 Available return code.
This return code is not currently used and is available for future use.

3056 The LuMap LuGroup does not contain assigned LU.
The LuGroup assigned to the connection at this time does not contain the LU name that was already assigned to the connection. This is probably caused by having multiple LuMap statements for the same Client Identifier, which can change based on the Application name chosen.

3057 The Appl LuGroup does not contain assigned LU.
The LUs assigned to the chosen application do not match the LU name that was already assigned to the connection. This is probably caused by having LUs listed in the LuMap LuGroup that are not listed in the LuGroup (or LU list) assigned to the application.

3058 An invalid range was specified.
The range specified during storage cleanup was not found. The most probable cause is storage corruption. Contact the IBM Software Support Center.

3059 No ParmsGroup defined for PMAP on LU/PRTMAP.
An LUMAP or PRTMAP statement specified an associated ParmsGroup using the PMAP parameter. The associated ParmsGroup name could not be found. The associated ParmsGroup must be defined prior to the mapping statement.

305A The password was successfully changed.
The Telnet solicitor panel was used to change an existing password. The change was successful.

305B The requested LU is kept for another client.
The LU name requested cannot be used at this time because it is being kept for another client that has previously used the LU name. When the KEEPLU time expires, the LU will be available to other clients.

305C Earlier ParmsGroup map replaced with this one.
The same ParmsGroup was mapped to the same Client Identifier more than once. The last mapping is used to concatenate the parameter values.

4001 Available return code.
This return code is not currently used and is available for future use.

4002 TN3270E header is in error.
The TN3270E header in the message received from the client has an error. Using a client trace, analyze the header. If it appears the header is correct, contact the IBM Software Support Center.

4003 SSCP LU data is invalid.
Telnet received 3270 data, a response, or is still in session when the connection is in SSCP-LU mode. Using a client trace or DEBUG TRACE, verify that the client does not send 3270 data or a response after issuing a SYSREQ to change to SSCP-LU mode.

4004 TN3270E subfunction was not negotiated.
The function requested by the client in the TN3270E header was not negotiated to be supported during connection startup. Using a client trace or DEBUG TRACE, determine the unsupported function and why the client is requesting function that is not supported.

4005 TN3270E datatype is not supported.
Telnet does not accept BIND, UNBIND, or NVT data from the client. Determine why the client is sending this data.

4006 Data received from the client is invalid.
Probable causes include receiving a TN3270E header with no data or receiving a response with an invalid flag value. Refer to RFC 2355 for valid response values. Using a client trace or DEBUG TRACE, determine what data is invalid and why the client is sending this data. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for information about accessing RFCs.

4007 VTAM Rsp received but previous rsp not complete.
A response was received from VTAM before an earlier response was completed. Contact the IBM Software Support Center.

4008 VTAM Rsp received but was not expected.
A response was received from VTAM but Telnet was not expecting a response. Contact the IBM Software Support Center. A VTAM internal trace in addition to the Telnet debug information will be needed.

4009 Negative VTAM Rsp is invalid.
A negative response from VTAM was received with invalid sense information according to RFC 2355. Determine why the application sent the invalid sense information and change the application. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for information about accessing RFCs.

400A Printer data is invalid.
Telnet received either SCS data or 3270 data. Whichever datatype was received is not supported by the client according to connection negotiation. Check the client to see if the data type option can be turned on or off. Otherwise, use a client trace or DEBUG TRACE during connection setup to verify what options are supported.

400B No data to send to the client.
An RPL was received from VTAM. Telnet determined that no data is available to send to the client. Contact the IBM Software Support Center.

400C BIND being sent to the client is invalid.
Probable causes are a BIND was already sent to the client or the BIND is zero length. Verify that the application is sending the correct sequence of BINDs/UNBINDs. If the application does not appear to be at fault, contact the IBM Software Support Center.

400D UNBIND being sent to the client is invalid.
Probable causes are an UNBIND was already sent to the client, a BIND was never sent, or the UNBIND is zero length. Verify that the application is sending the correct sequence of BINDs/UNBINDs. If the application does not appear to be at fault, contact the IBM Software Support Center.

400E Attempt to send BIND to client in SSCP-LU mode.
A request to send a BIND to the client is refused because the connection is in SSCP-LU mode. The client cannot accept binds.

400F Amount of data exceeded MAXRECEIVE value.
The amount of data received without an End Of Record indicator exceeded the value coded on the MAXRECEIVE statement or the default value. A probable cause is a broken client in a send loop or using a corrupted data length that is very large.

4010 Number of data packets exceeded MAXVTAMSENDQ.
A data packet in Telnet is created when an End Of Record indicator is received. At that time, the data packet is sent to host application or queued if the application cannot accept the data. In this case, the queue count exceeded the value coded on the MAXVTAMSENDQ statement or the default value. A probable cause is a hung application not receiving data. Determine why the host application is not receiving data.

4011 Negative response from client received.
This special case occurs when the original BIND sent to the client does not allow exception responses. Most clients require that a BIND received allows exception responses. To avoid numerous connection drops, Telnet adds exception response to the bind if not already specified. In this case an exception response was returned from the client. Telnet knows the application is not able to handle the exception and ends the session. It must be determined why the client found exception with the data it received.

4012 Invalid Send to VTAM negotiation incomplete.
This special case occurs when a VTAM send is attempted prior to negotiation completion. The client is usually a line mode client that sends in a carriage return or line feed before the negotiation is complete. The client will be disconnected.

4013 SNA sense error.
SNA Sense data was expected in the data from the client, but the data length was not long enough to contain the SNA Sense code.

4014 Client negotiation loop detected.
This special case occurs when a given client loops sending the same negotiation command to the Telnet Server. The client will be disconnected. Use a client trace or DEBUG TRACE to identify the repeated command and determine why the client is in a negotiation loop.

5001 Invalid TN3270E function code while negotiating.
The client is requesting an invalid TN3270E function code during function negotiation. Change the client so it does not request the invalid function. Using a client trace or DEBUG TRACE, determine what invalid function codes are being requested.

5002 Invalid TN3270E function during negotiation.
The client is requesting an invalid TN3270E function during function negotiation. Using a client trace, determine what invalid function is being requested.

5003 Printer negotiation does not allow SCS or DATA.
During connection negotiation with a printer, neither SCS nor DATA datatypes were negotiated. The client must support at least one of these datatypes to accept printer data from Telnet.

5004 WILL or DO command request rejected.
A Telnet command request from the client is unknown to the server and will be rejected. PARM1 contains either the WILL (X'FB') or the DO (X'FD') and PARM2 contains the command.

5005 End-of-Record negotiation option failed.
A failure occurred during negotiation. Contact the IBM Software Support Center.

5006 Transmit Binary negotiation option failed.
A failure occurred during negotiation. Contact the IBM Software Support Center.

5007 Terminal being taken over is inactive.
A probable cause is that the original LU is inactivated. Issue an INACTLUS display to confirm that the LU is inactive.

5008 Available return code.
This return code is not currently used and is available for future use.

5009 An unknown negotiation error was detected.
A negotiation error was detected by Telnet but is not a known error type. Use a client trace to determine what negotiation command is in error.

500A An unexpected new environment command detected.
An unexpected command for the new environment function was received from the client. Use a client trace or DEBUG TRACE to determine what negotiation command is in error.

6001 SSL/TLS failure while getting client ID.
Get ClientID request failed during SSL processing. Parm1, Parm2, and Parm3 are set to return value, return code, and reason code from the get clientid request respectively. Contact the IBM Software Support Center.

6002 SSL/TLS handshake failed.
SSL handshake with client failed. Parm1 contains the error code and Parm3 contains the function. Error codes the customer might be able to correct are:

108 The System SSL library (SGSKLOAD) could not be loaded. Ensure that TCP has access to this library.

401 Certificate expired.

402 None of the encryption algorithms requested by the client are supported for this port. If the ENCRYPT block is coded in the TELNETPARMS block, verify all desired algorithms are included. Ensure the correct level of client code is installed on the client.

403 Valid server certificate not found.

404 Bad certificate.

405 The certificate type is not supported.

410 Bad message. The following are some situations that might cause this error:
- If using client authentication, the client certificate was rejected during the SSL handshake. Possible problems are: the certificate expired, the certificate is not issued by a trusted CA, the certificate is in the Certificate Revocation List (CRL).
- If using client authentication, this message might occur if the client does not have the certificate immediately available. The client will reconnect when the client certificate is available.

420 Connection closed by peer. If using client authentication, some clients disconnect when the server requests the client certificate and will reconnect when the client certificate is available.

428 No key found for server certificate.

For other errors, contact the IBM Software Support Center.

6003 SSL/TLS client authentication failed.
Client Authentication requested but client did not provide a valid certificate. Either the client did not provide a certificate or the server does not consider the client certificate valid. The certificate of the CA that issued the client certificate must be in the keyring of the server and must be Trusted. Also ensure that the client certificate is not expired. Parm1 might contain the system security return code for the handshake. Parm3 might contain the function being processed when the error occurred.

6004 SSL/TLS initialization failed. No Telnet security task.
The system security interface task is not initialized. Look for an earlier DEBUG message to see why the system security initialization task failed.

6005 SSL/TLS READ failed.
An error occurred during a read. Parm1 contains the return code. Contact the IBM Software Support Center.

6006 SSL/TLS Give Socket failed.
The Give Socket process failed during the SSL/TLS handshake. Parm1, Parm2, and Parm3 contain the return value, return code, and reason code from the Give request respectively. Contact the IBM Software Support Center.

6007 SSL/TLS Take Socket failed.
The Take Socket process failed during the SSL/TLS handshake. Parm1, Parm2, and Parm3 contain the return value, return code, and reason code from the Take request respectively. Contact the IBM Software Support Center.

6008 SSL/TLS SEND failed.
An error occurred during a send request. Parm1 contains the return value and Parm2 contains the return code. Parm3 might contain the reason code. Contact the IBM Software Support Center.

6009 SSL/TLS userid mismatch.
On takeover reconnect, the user ID associated with the client certificate mismatched the original value.

600A SSL/TLS client authentication mismatch.
Takeover reconnect was attempted with a reduced CLIENTAUTH level. Ensure the correct level of client code is installed on the takeover client.

600B SSL/TLS invalid negotiation subcommand.
During negotiated SSL/TLS, a subcommand other than StartTLS_Follows was received. Protocol used by the client might not match that used by the server. Use a client trace or DEBUG TRACE to verify that the client is using the correct SSL negotiation.

600C SSL/TLS unexpected negotiation command.
A StartTLS_Follows subcommand was received but negotiated SSL/TLS had not been requested by the server. Protocol used by client might not match that used by the server. Verify that the client is using the correct security negotiation.

600D Negotiated SSL/TLS rejected by client.
Negotiated SSL/TLS is rejected by the client. The Telnet server attempted to negotiate SSL/TLS but the client responded WONT STARTTLS. The CONNTYPE for this connection is SECURE or NEGTSECURE. Since a secure connection is required and the client did not indicate a desire to enter an SSL connection, the connection is closed.

600E SSL/TLS handshake timed out.
The time required for the client to respond to the handshake request exceeded the value coded on the SSLTIMEOUT statement or the default. This might be expected if CONNTYPE SECURE is specified for the connection and the client uses the negotiated security protocol. Otherwise, increase the time to wait or determine why the client is not responding quickly enough.

600F System SSL interface initialization failed.
The setup of the security interface failed. Secure ports will not be available, but basic Telnet is available. The Telnet interface task was set up, code began to run, but an error occurred during initialization. Parm1 contains the security task return code. If the reason code you receive is not listed or you are unable to determine the cause of the failure, refer to the *z/OS System Secure Sockets Layer Programming*. These codes are passed to Telnet by System SSL. If Parm1 is one of the following values try the appropriate action before calling the IBM Software Support Center. The keyring file name is case sensitive. When adding the keyring name to the Telnet profile, be sure that the correct case is used.

- 102 – Keyring file I/O error
  - Unable to read the keyring file required for secure communications.
  - Secure communications cannot continue.
  - Verify that the hfs file system is operating correctly.
  - Ensure that the file permissions on the keyring file are correct.
- 202 – Keyring file open failed
  - Unable to open the keyring file or the keyring's password stash file, which are required for secure communications. If CRLLDAPSERVER is specified, the LDAP server might not be accessible.
  - Secure communications cannot continue.
  - Verify that the hfs file system is operating correctly.
  - If an hfs keyring is used, verify the stash file is in the same directory as the keyring file.
  - If the keyring file is an MVS data set, verify that the stash file is also an MVS data set.
  - Ensure that the file permissions on the keyring file are correct.
  - Verify that the LDAP server, if specified, can be accessed from this stack.

- 103 – Keyring file bad format
  - The keyring file format is incorrect.
  - Secure communications cannot continue.
  - Ensure the keyring file is correct.
- 201 or 408 – Keyring file bad password
  - The password is incorrect or no stash file is found.
  - This error can also occur if crypto hardware and ICSF are active but the TCP user ID was not given access to the RACF CSFSERV class resources.
  - Ensure that a stash password file was created. If the password was changed recently, ensure the stash file was re-created. If crypto hardware is installed, ensure TCP was permitted to the RACF CSFSERV resources.
- 401 – the default certificate in the keyring file has expired or is outside the valid date range. The default certificate in the keyring file is no longer valid. Refresh the certificate in the keyring file.
- 428 – No private key. The private key for the server certificate is not in the keyring file. Ensure the keyring contains both the server certificate and private key.

6010 Attach of the security interface task failed.
The MVS macro, ATTACH, failed to attach the SSL task during Telnet initialization. Parm1 contains the ATTACH return code. Contact the IBM Software Support Center.

6011 The security interface task has ended in error.
The security interface task ended because of an error condition. Instance 01 indicates that the task was set up properly and at some later time, an error occurred. Parm1 contains the interface task return code. Instance 02 indicates that the task was set up, but the interface task code never ran. Parm1 contains a system completion code. Or it indicates that the task was set up, the interface task code was initialized, but an error quickly occurred. Parm1 contains the interface task return code. Contact the IBM Software Support Center.

6012 ISTENINI could not be loaded.
The ISTENINI module used by TN3270 security support could not be loaded. Parm1 contains the return code from the LOAD. Ensure ISTENINI is accessible and APF authorized. Secure ports will not come on line. Basic ports are not affected.

6013 CEEPIPI environments could not be initialized.
The CEEPIPI environment used by TN3270 security support could not be initialized. Parm1 contains an additional internal return code useful to IBM Support. Secure ports will not come on line. Basic ports are not affected. Contact the IBM Software Support Center.

6014 CEEPIPI environments could not be expanded.
The CEEPIPI environment used by TN3270 security support could not be expanded. Parm1 contains an additional internal return code useful to IBM Support. There are more encryption requests than the current C environments created by Telnet can handle. Processing will continue, but response time might be degraded. The most probable reason additional CEEPIPI environments could not be increased is that the required storage is not available.

6015 SSL/TLS HANDSHAKE WRITE did not complete immediately
The SSL/TLS WRITE for HANDSHAKE is issued asynchronously. Since this process is running in TCB mode, a SUSPEND cannot be issued without blocking all other HANDSHAKES. Ensure that the stack buffer size is at least 1024 bytes to prevent this. The session setup will fail and disconnect the client.

6016 Passticket could not be obtained.
An attempt to obtain a Passticket from RACF failed for an Express Logon macro. PARM1 contains the SafRC from RACF if it is available. Contact the IBM Support Center.

6017 Available return code.
This return code is not currently used and is available for future use.

6018 Available return code.
This return code is not currently used and is available for future use.

6019 Available return code.
This return code is not currently used and is available for future use.

601A Available return code.
This return code is not currently used and is available for future use.

601B Available return code.
This return code is not currently used and is available for future use.

601C Available return code.
This return code is not currently used and is available for future use.

601D Available return code.
This return code is not currently used and is available for future use.

601E Available return code.
This return code is not currently used and is available for future use.

601F KEYRING is required on a Secure Port.
A Secure Port was defined in TelnetParms but KEYRING was not specified in either TelnetGlobals or the TelnetParms statement.

6020 The KEYRING name is invalid.
The KEYRING name is invalid. The name might be different from the name used with current active profiles. Use the Profile Display command to see if a Keyring is currently defined. The name might be invalid because it is different from one already accepted for the Profile being processed.

6021 The SSL parameter is invalid on non-Secure Port.
SSL parameters are not valid on Basic Ports. It is assumed that the port was meant to be secure because of the SSL parameter specified. If it is intended to be basic, remove the SSL parameter.

6022 The SSL Encryption value is invalid.
The SSL Encryption value contained in Parm3 is not valid compared to the values supported by the SSL subsystem. These values can be determined by displaying the Telnet defaults using the Object display and specifying ID=*DEFAULTS.

6023 The Encryption statement has no valid values.
All values specified are invalid or no values were specified. At least one valid value must be specified to allow successful processing of the profile.

6024 KEYRING type SAF specified but SAF unavailable.
The KEYRING statement specified a SAF name but the Secure Access Facility is not available. Ensure the SAF product (such as RACF) is available and reprocess the profile.

6025 ClientAuth SAF specified but SAF unavailable.
The ClientAuth statement specified SAFcert but the Secure Access Facility is not available. Ensure the SAF product (such as RACF) is available and reprocess the profile.

6026 ClientAuth SAF specified but SAFcert unavailable
The Client Authentication portion of the Secure Access Facility is not available. Basic SAF might be available even when the Client Authentication portion is not available. Ensure the Client Authentication portion is available and reprocess the profile.

6027 Global SSL Parm ignored on basic port.
Global SSL parameters cannot sift down to Basic ports. To avoid this message when a mixture of Basic and Secure ports are defined, specify the SSL parameters in the TelnetParms blocks instead of in TelnetGlobals.

6028 ParmsGroup SSL Parm ignored on basic port.
SSL parameters have no affect on Basic ports and are ignored in ParmsGroup statements. If the BeginVtam block defines both Basic and Secure ports the message can be avoided by splitting the BeginVtam block into two blocks, one for Basic and one for Secure. If the BeginVtam block defines only Basic ports, remove the SSL parameters.

6029 The CRL LDAP SERVER name is invalid.
The CRL LDAP Server name is invalid because a CRL LDAP Server with a different name was already defined for active profiles. To change the name, inactivate all Secure ports and then process the new profile with the new CRL LDAP Server name.

602A The CRL LDAP does not have a keyring.
The CRL LDAP Server is used with secure connections, which requires the specification of a keyring. To use the CRL LDAP Server, process a new profile with a valid keyring.

602B The SSL task initialization failed.
The Telnet task that supports SSL did not initialize. Parm1 contains the initialization return code. This should not occur. Contact the IBM Support Center.

7001 Invalid character entered on logon panel.
After translating input to upper case a character is determined to be invalid. Refer to PARM1 for the hex value of the invalid character.

7002 Load of the default USS table failed.
The default USS table could not be loaded. The most probable reason is that it is not part of the system library accessible to Telnet. Parm1 might contain the system completion code and Parm2 might contain the reason code. If the load module is accessible to Telnet, contact the IBM Software Support Center.

7003 The default USS table header is an invalid type.
The USS table must be assembled with FORMAT=DYNAMIC using VTAM macros from V4R1 or newer. The USS table must have a control block ID of BD to be valid. If the IBM default USS table is being used, contact the IBM Software Support Center.

7004 Storage request for USS/INTERPRET/LUEXIT table failed.
Storage needed to manage or save the table was not obtained. Increase storage availability. Parm1 contains the hex amount of storage requested.

7005 The MVS LOAD of the USS/INTERPRET/LUEXIT failed.
The LOAD of the table/exit failed. Parm1 contains the reason code of the load failure.

7006 The MVS BLDL of the USS/INTERPRET/LUEXIT table failed.
The specified table/exit was not found. Parm1 contains the value returned by the MVS BLDL macro in R15. See *z/OS DFSMS Macro Instructions for Data Sets* for information about the MVS BLDL macro. This probably occurred because the table is not in a data set accessible by TCP/IP.

7007 The internal USS table type is invalid.
The table being loaded does not have the correct table identifier of BD or it was not assembled with FORMAT=DYNAMIC using VTAM macros from V4R1 or newer.

7008 The internal INTERPRET table type is invalid.
The table being loaded does not have the correct table identifier of BE. The table is not an INTERPRET table.

7009 No sequence match of input by interpret table.
Input data was passed to the Interpret table but did not match any of the sequences in the table. This is a valid situation and the input data will be processed by the USS table.

700A The returned NQN is in an invalid format.
The Interpret table exit routine returned a network qualified name with an invalid format. Valid format is name.network where name and network are each a maximum of eight characters.

700B The LU/PRT exit failed the function call.
The LU/PRT exit was invoked with a specific function call. The exit failed the request with a nonzero return code. In the case where the function is ASSIGNLU, the client connection request is terminated. Parm1 contains the return code from the LU exit. Parm2 contains the specific function call. Parm3 contains the name of the LU exit. If this error is unexpected, the LU exit should be investigated to determine the reason for rejecting the requested function.

700C The LU/PRT exit input parameters were changed.
The parameters for an LU or PRT group exit were changed. Once the exit is activated, the input parameters are not allowed to be changed. This error was detected during VARY TCPIP,,OBEYFILE file processing.

700D The LU/PRT exit has been disabled due to abends.
The LU or PRT exit was disabled. It is disabled because the exit exceeded the maximum number of abends allowed for a user exit. Currently this abend threshold is set at a maximum of three abends within a 10-minute period. The root cause of the LU/PRT exit abends should be investigated and corrected.

700E Calling LU/PRT exit.
The Debug Trace option was selected causing trace messages to occur. This debug message displays the contents of the parameter list being passed to the LU/PRT exit at invocation time.

700F Return from LU/PRT exit.
The Debug Trace option was selected, causing trace messages to occur. This debug message displays the contents of the parameter list upon return from the LU/PRT exit invocation.

7010 LU/PRT exit not allowed with associated printer.
When an associated printer is coded on an LUMAP statement, neither the LU group or PRT group is allowed to be defined as an exit. The LUMAP statement is ignored.

8001 Available return code.
This return code is not currently used and is available for future use.

8002 Configuration task setup failed.
The setup of the Telnet configuration task failed. Without the configuration task, Telnet cannot receive any profile statements or operator commands. Parm1 contains the configuration task return code. Contact the IBM Software Support Center.

8003 Attach of the Configuration task failed.
The MVS macro, ATTACH, failed to attach the Configuration task during Telnet initialization. Without the configuration task, Telnet cannot receive any profile statements or operator commands. Parm1 contains the ATTACH return code. Contact the IBM Software Support Center.

8004 The Configuration task has ended in error.
The Configuration task ended because of an error condition that caused three abends within a 10-minute period. Without the configuration task, Telnet cannot receive any profile statements or operator commands. Contact the IBM Software Support Center.

8005 Available return code.
This return code is not currently used and is available for future use.

8006 Incomplete profile update aborted for this one.
An earlier profile process that did not finish because of an error being aborted without being applied. The new profile request is processed. It is uncommon to abort a profile in progress. If the problem persists, contact the IBM Support Center.

8007 No IP mask exists for the delete request.
The IP subnet mask entry is being deleted but the IP mask cannot be found. Data corruption is the probable cause. If the problem persists, contact the IBM Support Center.

8008 An AbendTrap has already been set.
A VARY AbendTrap command was already issued setting the Abend Trap. Use the Profile display command to see what is set. If you want to change the trap, turn off the current trap by specifying ″OFF″ as the module name and then set the new trap.

8009 Unknown display request code.
Internal processing created an incorrect display request code. This should not occur. If the problem persists, contact the IBM Support Center.

800A Invalid Profile specified on Display command.
The profile number or type specified on the DISPLAY command is invalid. Issue the command with a valid profile number or no profile number.

800B Invalid DEBUG command option.
A DEBUG option other than OFF was specified. OFF is the only valid command option at this time.

800C Ending profile processing but none in progress.
Telnet received an ″end profile″ command from the TCP/IP Configuration task but no profile process was in progress. This should not occur. If the problem persists, call the IBM Support Center.

800D No port match for VARY command.
The port number, range, or type specified does not match any active port. reissue the command with an active port.

800E No port specified but multiple ports exist.
The VARY command requires that a port number, range, or type be specified if more than one port is active. Without a port specified, it is unclear whether all ports were meant to be affected.

800F No ports active for VARY command.
There are no ports active in Telnet at this time. The command has no affect.

8010 Unknown profile statement SubType encountered.
Internal processing created an unknown profile statement SubType. Parm1 contains the invalid subtype. This should not occur. If the problem persists, contact the IBM Support Center.

8011 Transform activation request ignored.
Transform is already active on another port and cannot be activated on this port. The original port must be deactivated before transform can be specified on this port.

8012 Invalid length received for profile statement.
An internal error caused the statement record to indicate an incorrect length for the profile statement received from the TCP/IP stack. If the problem persists, contact the IBM Service Center.

8013 Invalid parameter received in profile statement.
An internal error caused the statement record to indicate an incorrect parameter for the profile statement received from the TCP/IP stack. If the problem persists, contact the IBM Service Center.

8014 LUSESSIONPEND replaces QUEUESESSION.
LUSESSIONPEND and QUEUESESSION are mutually exclusive. If both are specified at any time in the profile, LUSESSIONPEND is used.

8015 QUEUESESSION ignored, already LUSESSIONPEND.
LUSESSIONPEND and QUEUESESSION are mutually exclusive. If both are specified at any time in the profile, LUSESSIONPEND is used.

8016 Invalid devicetype specified.
An invalid device type was specified on the TELNETDEVICE statement. See the TELNETDEVICE statement in the *z/OS Communications Server: IP Configuration Reference* for a list of valid device types.

8017 Invalid 3270E devicetype specified.
An invalid 3270E device type was specified on the TELNETDEVICE statement. See the TELNETDEVICE statement in the *z/OS Communications Server: IP Configuration Reference* for a list of valid device types.

8018 Codepage setup, including defaults, failed.
No translation tables from the specified Codepage statement or the defaults were generated. The internal Telnet translation tables will be used.

8019 Unknown profile statement ReqType encountered.
Internal processing created an unknown profile statement ReqType. Parm1 contains the invalid ReqType. This should not occur. If the problem persists, contact the IBM Service Center.

801A BEGINVTAM block with no port number.
Multiple BEGINVTAM blocks were specified in the profile. At least one block did specify a port number implying multiple ports. It is unclear which port the no-port BEGINVTAM should be matched with and is therefore ignored.

801B Multiple BEGINVTAM blocks. Last one is used.
Multiple BEGINVTAM blocks for the same port were found in the profile. The last block is used.

801C Multiple TELNETPARMS blocks. Last one is used.
Multiple TELNETPARMS blocks for the same port were found in the profile. The last block is used.

801D Multiple PORT blocks. Last one is used.
Multiple TELNETPARMS blocks for the same port were created from multiple PORT statements. The last block is used.

801E BEGINVTAM block with no TELNETPARMS block.
A BEGINVTAM block was found in the profile but has no matching TELNETPARMS block. A port definition must have both a BEGINVTAM block and a TELNETPARMS block to be successfully started or updated.

801F TELNETPARMS block with no BEGINVTAM block.
A TELNETPARMS block was found in the profile but has no matching BEGINVTAM block. A port definition must have both a BEGINVTAM block and a TELNETPARMS block to be successfully started or updated.

8020 Initialization of the Telnet Port failed.
Port activation includes attaching a load module that performs all the connection tasks. The attach or initialization of the task failed. The return code is contained in Parm1.

8021 Maximum number of Telnet Ports exceeded.
The maximum number of Telnet ports is 255. No more ports can be activated until existing ports are inactivated.

8022 Port has both secure and basic components.
The port being quiesced, resumed or stopped has both secure and basic components. Because the port is both secure and basic, the command will not alter the port. To alter the port, reissue the command for the port without the secure or basic option.

8023 Placement accepted. Use TELNETPARMS in future.
The parameter placement in BEGINVTAM is accepted at this time but will have to be moved to TELNETPARMS in

a future release. Move the parameter now to stop receiving the warning message. The parameter statement can now be placed in the TELNETGLOBALS, TELNETPARMS, or ParmsGroup.

8024 Display syntax obsolete. Use OBJ/CLID display.
The display command is no longer supported in its original format. The use is accepted but is internally translated to one of the OBJect or CLient IDentifier display commands. Use the appropriate OBJect or CLient IDentifier display command to avoid this message.

8025 The LU/PRT Group must contain < 4294967296 LUs.
The number of LUs in the group exceeds the Telnet limit of 4 294 967 296. Reduce the number of LUs in the group.

8026 Usage accepted but obsolete in future release.
The statement usage is accepted at this time but will become obsolete in a future release.

- INTERNALCLIENTPARMS - Replace with TELNETPARMS.
- QUEUESESSION - For the DEFAULTAPPLs that QUEUESESSION is affecting, add an ALLOWAPPL statement with the QSESSION option.

8027 Display syntax obsolete. Use PROF,det display.
The display command is no longer supported in its original format. The use is accepted but is internally translated to the Profile detail display command. Use that display to avoid this message.

8028 First character must be equal and not numeric.
The old style LU range requires that the first character be alphabetic or a national character (@#$) and the low entry first character must match the high entry first character. The first character may be an alphabetic range using the range rules third parameter.

8029 Variable must be all numeric or all alpha.
The old style LU range requires that the variable portion be entirely alphabetic or entirely numeric. If a combination of alphabetic and numeric is desired, include the range rules.

802A First entry is higher than the second entry.
The first entry must be lower than the second entry to create a valid range. Ensure that the first variable column in the first entry is lower than the first variable column in the second entry. The order for Telnet LU range characters is 0-9, A-Z, @, #, $.

802B Port stop in progress for Obeyfile
The VARY TCPIP,,OBEYFILE processing found that the port for TELNETPARMS in in the process of stopping. Reissue the command after stop is complete.

802C Variant does not match range entries.
The variant is not valid for the LU range entries specified. The range might indicate Fixed when the Start character is different from the End character or the character might not be valid for the variant specified.

802D The variant must be fixed.
The variant must be fixed for the character index. A character is the same in both the Start and End names with a variant other than Fixed with no variable character to the left.

???? Return code explanation not available.
A message description for the specified return code has not been written. Contact the IBM Software Support Center.

System Action: None.

User or Operator Response: See above for specific operator response.

System Programmer Response: See above for specific system programmer response.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6038I TELNET COMMAND *command name* **COMPLETE**

Explanation: The requested command completed successfully.

command is one of the following:

INACT The INACT request is used to inactivate a Telnet LU name so Telnet will not use the name. The LU as managed by Telnet was successfully inactivated.

ACT The ACT request is used to reactivate a Telnet LU name so Telnet will be able to use the name. The LU as managed by Telnet was successfully activated.

DEBUG
The DEBUG request is used to change the debug status of all active connections. The debug option for all connections was successfully turned off.

ABENDTRAP
The ABENDTRAP request is used to set up an abend if the specified module detects an error. Return code and instance are optional parameters to restrict the trap. The module and optional return code/instance number have been successfully set.

QUIESCE
The QUIESCE request is used to block any additional connections from being accepted. The port(s) requested have been found and processing started.

RESUME
The RESUME request is used to unblock a port that is blocked because of an earlier QUIESCE command or quiesced automatically because of an ACCEPT error. The port(s) requested have been found and processing started.

STOP The STOP request is used to stop ports. Any active connections are cleaned up. The port(s) requested have been found and processing started.

name is either the LU name being activated or inactivated, the debug level requested (Off is currently the only option), or the module name specified on the Abendtrap command, depending on the command. Quiesce, Resume, and Stop do not use this field.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

EZZ6039I TELNET COMMAND *command name* **FAILED, RCODE=***rcode*

Explanation: The request did not complete successfully.

command is one of the following:

INACT The INACT request is used to inactivate a Telnet LU name so Telnet will not use the name.

ACT The ACT request is used to reactivate a Telnet LU name so Telnet will be able to use the name.

DEBUG
The DEBUG request is used to change the debug status of all active connections.

ABENDTRAP
The ABENDTRAP request is used to set up an abend if the specified module detects an error. Return code and instance are optional parameters to restrict the trap.

QUIESCE
The QUIESCE request is used to block any additional connections from being accepted. The command request will fail if there is no port match or no port was specified when multiple ports are active.

RESUME
The RESUME request is used to unblock a port that is blocked because of an earlier QUIESCE command or quiesced automatically because of an ACCEPT error. The command request will fail if there is no port match or no port was specified when multiple ports are active.

STOP The STOP request is used to stop ports. Any active connections are cleaned up. The command request will fail if there is no port match or no port was specified when multiple ports are active.

name is either the LU name being activated or inactivated, the debug level requested (Off is currently the only option), or the module name specified on the Abendtrap command, depending on the command.

rcode is the return code of the failure. The return codes are documented under message EZZ6035I.

System Action: None.

User or Operator Response:

For the INACT request, verify that the LU is not already inactivated by using the Telnet INACTLUS display. The LU does not have to already exist in Telnet definitions to be inactivated. It should fail only if it is already inactivated.

For the ACT request, verify that the LU is inactivated by using the Telnet INACTLUS display.

For the DEBUG request, verify that the option is valid. At this time, OFF is the only valid option.

For the ABENDTRAP request, verify that the trap is not already set by using the Telnet PROFILE display.

For the QUIESCE, RESUME, and STOP commands ensure that a port number or range includes an active port. If no port number was specified, ensure that only one port is active.

System Programmer Response: None.

Source Data Set: EZAZMTNS

EZZ6040I TELNET *action* **FAILED FOR PORT** *pnum*, **RCODE=***rcode*

Explanation: A Telnet Server component encountered a severe error.

action is one of the following:

WLM PROCESSING
The Telnet Server encountered a severe error in the WLM component.

PROFILE UPDATE
The Profile has a severe error and will not be activated. The existing profile remains.

pnum is the port number with the failing action.

rcode is the return code. The code identifies the error found during processing. See message EZZ6035I for a detailed description of the return code.

System Action: None.

User or Operator Response: Contact the system programmer to diagnose this error.

System Programmer Response: None.

Source Data Set: EZAZMTNS

EZZ6050I *prefix port portstate* **PROF:** *profnum* **CONNS:** *numconns*

Message Format:

```
----- PORT:     port   port_stat     PROF: CURR CONNS:      0
----- QUAL: qualification
```

Explanation: *port* is the Port number for the preceding output lines.

portstate is one of the following:

QUIESCED
The port is in a quiesced state at the time of the display. No new connections can be accepted while quiesced.

ACTIVE
The port is in an active state at the time of the display. New connections can be accepted.

profnum is the profile number for data just displayed.

numconns is the number of connections associated with this profile.

qualification indicates that the profile information being displayed is associated with a qualified port. The port qualification might display as an IP address (xxx.xxx.xxx.xxx), or as a linkname up to sixteen characters or the word NONE to indicate that the port is not defined with a qualifier.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

EZZ6052I *displayedlines* **OF** *totallines* **RECORDS DISPLAYED**

Explanation: Specifies the number of data lines displayed and the total lines possible to display for every Telnet display command.

displayedlines is the number of data lines displayed.

totallines is the total number of data lines possible.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6057I NO QUALIFYING MATCHES

Explanation: Specifies that no matching data for the display query was found.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6060I TELNET PROFILE DISPLAY

Message Format:

```
 PERSIS  FUNCTION   DIA  SECURE   TIMERS  SMF   MAX  LINE
(LMTQAK)(OATSQSWHT)(DRF)(PCKLECX)(IKPSTS)(ITIT)(RSQ)(BSCTT)
 ------  ---------  ---  -------  ------  ----  ---  -----
 persis  function   diag secure   timers  smf   max  line
   ABENDTRAP     :modname  rcode instance  trap_status
```

Explanation: The PROFILE Display command.

Profile and port divider line information follows each profile/port section. See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details.

If a Secure port exists, KEYRING and optional CRL LDAP Server information is displayed after the profile summary data.

persis represents different persistence-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) L if LuSessionPend coded.

Position(2) M if MSG07 coded.

Position(3) T if TkoSpecLu, R if TkoSpecLuRecon coded.

Position(4) Q if QueueSession coded.

Position(5) P if DropAssocPrinter coded.

Position(6) K if KeepLU coded.

function represents different general function options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) O if OldSolicitor coded.

Position(2) A if SingleAttn coded.

Position(3) T if TN3270E coded.

Position(4) S if SimClientLu coded.

Position(5) Q if SequentiaLU coded.

Position(6) S if SnaExt coded.

Position(7) W if WlmClusterName coded.

Position(8) H if Hostname lookup will be done.

Position(9) T if TelnetDevice coded.

diag represents different diagnostics-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) T if Debug Trace, D if Debug Detail, S if Debug Summary coded, E if Debug Exception coded.

Position(2) C if Debug routing is to the console, J if Debug routing is to the joblog.

Position(3) F if FullDataTrace coded.

secure represents different security-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) S if SecurePort coded.

Position(2) N if ConnType NegtSecure, B if ConnType Basic, S if ConnType Secure, A if Conntype Any, 0 if Conntype None coded.

Position(3) M if MVS keyring, H if HFS keyring, S if SAF keyring coded.

Position(4) L if Crl LDAP server coded.

Position(5) E if Encryption values coded, D if Default coded.

Position(6) S if Clientauth SSL, F if Clientauth SAF coded.

Position(7) X if ExpressLogon coded.

timers represents different timer-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) I if Inactive timer coded.

Position(2) K if KeepInactive timer coded.

Position(3) P if PrtInactive timer coded.

Position(4) S if ScanInterval timer coded.

Position(5) T if Timemark timer coded.

Position(6) S if SSLtimeout timer coded.

smf represents different SMF options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) I if SMFinit num coded.

Position(2) T if SMFterm num coded.

Position(3) I if SMFinit TYPE119 coded.

Position(4) T if SMFterm TYPE119 coded.

max represents different maximum limit related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) R if MaxReceive coded.

Position(2) S if MaxVtamSendQ coded.

Position(3) Q if MaxReqSess coded.

line represents different Linemode-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) B if BinaryLineMode coded.

Position(2) S if SGA coded.

Position(3) C if CodePage tables exist.

Position(4) T if DbcsTransform coded.

Position(5) T if DbcsTrace coded.

modname is the module name for abend trap.

rcode is the return code for abend trap if specified.

instance is the Instance for abend trap if specified.

trap_status is one of the following:

ACTIVE
The abend trap is active.

TRIPPED
The abend trap was tripped. The VARY ABENDTRAP command needs to be issued again to reactivate the trap.

OFF The abend trap was turned off. The VARY ABENDTRAP command needs to be issued again to reactivate the trap.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6061I TELNET INACTLUS DISPLAY

Message Format:

```
EZZ6061I TELNET INACTLUS DISPLAY
INACTIVE LUS
               luname    luname    luname    luname    luname
```

Explanation: The INACTLUS Display command. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details.

luname is the Name of inactive LU.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6064I TELNET CONNECTION DISPLAY

Message Format:

```
EZZ6064I TELNET CONNECTION DISPLAY
         EN                                          TSP
CONN     TY  IPADDR..PORT           LUNAME   APPLID   PTR LOGMODE
-------- --  ---------------------- -------- -------- --- --------
connid   en  ipaddr..port           luname   applid   tsp logmode
hostname
```

Explanation: The CONNECTION Summary Display command. Profile and port divider line information follows each profile/port section. See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details.

connid is the connection ID as it is known to TCPIP.

en is the encryption/hashing used, if SSL connection and is one of the following:

blank basic connection - no encryption or hashing used

NN SSL_NULL_NULL

NM SSL_NULL_MD5

NS SSL_NULL_SHA

4E SSL_RC4_MD5_EX

4M SSL_RC4_MD5

4S SSL_RC4_SHA

2E SSL_RC2_MD5_EX

DS SSL_DES_SHA

3S SSL_3DES_SHA

ipaddr..port is the client IP address and port number.

luname is the LU name that Telnet is using to represent the client to the VTAM appl.

applid is the application name with which the client is in session.

t is the terminal type of the client and is either:

T The client is a terminal.

P The client is a printer.

s is the session state of the connection and is one of the following:

A The connection has an active session with a VTAM appl

P The connection completed negotiation but not yet established a session with a VTAM appl

N The client and server are negotiating options to determine the protocol

p is the protocol type of the connection and is one of the following:

3 TN3270 mode

E TN3270E mode

L Line mode

T Transform mode

D DBCS transform mode

B Binary line mode

logmode is the logmode of the session.

hostname is the HostName of the client.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6065I TELNET CONNECTION DISPLAY

Message Format:

```
| EZZ6065I TELNET CONNECTION DISPLAY
|   CONN: connid               CLNTIP..PORT: ipaddr..port
|   LINKNAME: linkname         DESTIP..PORT: ipaddr..port
|   HOSTNAME: hostname
|   CONNECTED: time      date        STATUS: conn_status
|   PORT: port  QUAL: qualification
|     STATUS:  portstat portacc  ACCESS: connaccess
|   PROTOCOL: protocol LOGMODE: logmode  DEVICETYPE: devicetype
|     OPTIONS: options   3270E FUNCTIONS: e_functions
|                       NEWENV FUNCTIONS: ne_functions
|   USERIDS
|     RESTRICTAPPL: ruser     CLIENTAUTH: cuser
|     EXPRESSLOGON: euser
|   APPL: applname
|   LUNAME: luname    TYPE: termtype conntype
|   MAPS CONN IDENTIFIER      OBJECT   DEFAPPL        MAPOPTIONS
|     LU MAPPINGS:
|        ct clid            u object   defappl        mapoptions
|           APPL: applname  u object                  mapoptions
|     DEFAULTAPPL:
|        ct clid              nqobject
|     USS TABLE:
|        ct clid              object
|     INT TABLE: **N/A**
|     PARMS:
|   PERS  FUNCT    DIA  SECURE   TIMERS  SMF   MAX  LINE
|  (LMTQ)(OATSSWH)(DRF)(SCKLECX)(IKPSTS)(ITIT)(RSQ)(BDCTT)
|   ----  -------  ---  -------  ------  ----  ---  -----
|  persis function diag secure  timers   smf   max  line  source
```

```
|          ct clid
|   persis function diag secure  timers   smf   max  line  prmgrpname
|   persis function diag secure  timers   smf   max  line  FINAL
|   nn OF nn RECORDS DISPLAYED
```

Explanation: The CONNECTION DETAIL Display command. See message EZZ6081I for an explanation of the PARMS: fields.

connid is the connection ID as it is known to TCP/IP.

| *ipaddr..port* is the client host or destination IP address and port number.

linkname is the linkname, if any, of the connection.

hostname is the HostName of client.

time is the time at which the connection was made.

date is the date on which the connection was made.

conn_status is one of the following:

SESSION PENDING
The negotiation process is complete but a session is not established with an appl.

SESSION ACTIVE
The negotiation process is complete and a session is active with an appl.

NEGOTIATE IN PROGRESS
The negotiation process is not yet complete.

port is the port number for the following output lines.

| *qualification* indicates that the profile this connection is using is associated with a qualified port. The port qualification
| might display as an IP address (xxx.xxx.xxx.xxx), or as a linkname up to sixteen characters or the word NONE to
| indicate that the port is not defined with a qualifier.

portstat is the Port status and is either ACTIVE or QUIESCED.

portacc is the port access type and is either BASIC or SECURE.

connaccess is either Secure or Non-Secure. If Secure and the encryption type was negotiated, a two digit encryption type abbreviation follows. See EZZ6064I for a list of these values. If the access involves SSL or TLS the type and version will follow, such as, T1 for TLSV1 or S3 for SSLV3.

protocol is the connection protocol.

logmode is the logmode of the session.

devicetype is the device type that the client specified.

options are the connection options negotiated. See EZZ6064I for details.

e_functions are the 3270E Functions negotiated.

B Bind image to be sent to the client

S SysReq key function supported

R Response function supported

S SCS control codes supported (printer only)

D Data stream (3270) supported (printer only)

C Contention Resolution supported

N SNA Sense supported

ne_functions is the new Environment Function negotiated.

E Express Logon supported

ruser is the user ID used at the solicitor panel in response to the Telnet request for user ID/password because of Restrictappl being coded.

cuser is the user ID established from client certificate used during SSL handshake negotiation.

euser is the user ID established from client certificate used during new environment negotiation.

applname is the name of the application in session with the client.

luname is the LU name selected by Telnet to represent the client.

termtype is one of the following:

TERMINAL
The client is emulating a terminal

PRINTER
The client is emulating a printer

conntype is one of the following:

GENERIC
A generic connection was established.

SPECIFIC
A specific connection was established.

ct is the client Identifier type and is one of the following:

US User ID

HN Hostname

IP IP Address

UG User ID within a USERGROUP

HG Hostname within an HNGROUP

IG IP address within an IPGROUP

DI Destination IP address

LN Linkname

DG Destination IP address within a DESTIPGROUP

LG Linkname within a LINKGROUP

NL Null. No Client Identifier specified

clid is the client identifier used to map the object.

u is a greater than symbol (>), which identifies the LU group being used.

object is the object name being mapped.

defappl is the default appl on LUMAP statement.

mapoptions contains mapping options

Position(1) Set to S if Specific mapping, or G if Generic mapping.

Position(2) Set to L if LOGAPPL coded, or Q if QINIT coded.

Position(3) Set to F if FIRSTONLY coded.

Position(4) Set to D if DEFONLY coded.

Position(5) Set to K if KEEPOPEN coded.

na is **N/A** to indicate that no Object was mapped for this Object type

nqobject is the Network Qualified Object name being mapped.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6067I TELNET WLM DISPLAY

Message Format:

```
EZZ6067I TELNET WLM DISPLAY
WLM CLUSTER NAME          STATUS
------------------  ---------------------
wlm_name            status
```

Explanation: The WLM Display command. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details.

wlm name is the WLM name Telnet used to register with the DNS.

status is the current status of the registration. The status will be one of the following:

- Never Registered - The port was quiesced when this name was first used in a profile
- Registered - Telnet is currently registered by this name with the DNS
- DeRegistered - Telnet was registered by this name with the DNS but the port is currently quiesced
- Registration Failed - An error occurred while trying to register this name
- DeRegistration Failed - An error occurred while trying to deregister this name

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6080I TELNET PROFILE DISPLAY

Message Format:

```
 PERSIS  FUNCTION   DIA  SECURE   TIMERS  SMF   MAX  LINE
(LMTQAK)(OATSQSWHT)(DRF)(PCKLECX)(IKPSTS)(ITIT)(RSQ)(BSCTT)
 ------  ---------  ---  -------  ------  ----  ---  -----
 ******  **TSQ***T  EJ*  BB*****  ***STS  ****  RSQ  *SC**  *DEFAULT
 -M----  ---------  ---  -------  ------  ----  ---  -----  *TGLOBAL
 L-----  ---------  ---  -B-----  ------  ----  ---  -----  *TPARMS
 LM****  **TSQ***T  EJ*  BB*****  ***STS  ****  RSQ  *SC**  CURR
grp_id
actual_parm
```

Explanation: The PROFILE Display command.

Profile and port divider line information follows each profile/port section.

Figure 3.

See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details. See the *z/OS Communications Server: IP Configuration Reference* for an explanation of the parameters.

grp_id is the parameter group identifier used in the summary displays.

actual_parm is the parameter specified showing correct syntax.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6081I EZZ6081I TELNET CLIENTID DISPLAY

Message Format:

```
EZZ6081I TELNET CLIENTID DISPLAY
CLIENT ID          CONNS  OBJECT    OBJECT            ITEM
NAME               USING  TYPE      NAME     OPTIONS  MAPPED
------------------ ------ --------- -------- -------- --------
clid_type
  clid_name        clconn obj_type  obj_name map_opts da_name
clid_type: clid_name
      clid_entries
PARMS:
 PERSIS  FUNCTION   DIA  SECURE   TIMERS  SMF   MAX  LINE
(LMTQAK)(OATSQSWHT)(DRF)(PCKLECX)(IKPSTS)(ITIT)(RSQ)(BSCTT)
 ------  ---------  ---  -------  ------  ----  ---  -----
 persis  function  diag  secure   timers  smf   max  line   source
         ct clid
 persis  function  diag  secure   timers  smf   max  line  prmgrpname
         LU MAPPING maptype luname
 persis  function  diag  secure   timers  smf   max  line  prmgrpname
 persis  function  diag  secure   timers  smf   max  line  FINAL
 nn OF nn RECORDS DISPLAYED
```

Explanation: This message is displayed in response to the Telnet ClientID display command. It displays a detailed list of Client Identifier names and their mapped Objects sorted by Client Identifier, profile, and port. The CLIENTID display can be used to see what Client Identifiers are defined in the profile and details about the Client Identifier. The Client Identifier group details and parameter mapping results can be displayed when the ID= parameter is used in the display command.

Profile and port divider line information follows each profile/port section. See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details.

clid_type is the type of Client Identifier. Possible Client Identifier types are: -

USERID An exact user ID specified on a mapping statement.

HOSTNAME An exact hostname specified on a mapping statement.

IPADDR An exact IP address specified on a mapping statement.

USERGRP A user ID group name specified on a mapping statement.

HNGRP A hostname group name specified on a mapping statement.

IPGRP An IP address group name specified on a mapping statement.

DESTIP An exact destination IP address specified on a mapping statement.

LINKNAME An exact linkname specified on a mapping statement.

DESTIPGRP A destination IP address group name specified on a mapping statement.

LINKGRP A linkname group name specified on a mapping statement.

NULL The special case where no client identifier is specified.

clid_name is the name of the Client Identifier.

clid_entries is a list of Client Identifiers within a group.

clconn is the number of connections using the Client Identifier mapping statement.

obj_type is the type of Object. Possible Object types are:

DEFAPPL The application name specified on the DEFAULTAPPL statement.

LINAPPL The application name specified on the LINEMODEAPPL statement.

USS The USS table name specified on the USSTCP statement.

INT The interpret table name specified on the INTERPTCP statement.

LU The exact LU name specified on the LUMAP statement.

LUGRP The LUGROUP name specified on the LUMAP statement.

PRT The exact printer LU name specified on the PRTMAP statement.

PRTAPPL The application name specified on the PRTDEFAULTAPPL statement.

PRTGRP The PRTGROUP name specified on the PRTMAP statement.

PARMSGRP The PARMSGROUP name specified on the PARMSMAP statement.

obj_name is the name of the object being mapped on the mapping statement.

map_opts are additional mapping options on certain mapping statements. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option is not set or does not apply. Possible values by position are:

Position(1) Set to A or R indicating the LUMAP-DEFAPPL is Allowappl or Restrictappl, if DEFAPPL coded on the LUMAP statement.

Position(2) Set to L if LUs are mapped on the Allowappl or Restrictappl statement that matches the LUMAP-DEFAPPL application name coded on the LUMAP statement.

Position(3) Set to G or S indicating the mapping is Generic or Specific for the LUMAP or PRTMAP statement

Position(4) Set to D, A, or P indicating that the *da_name* is a Defappl, Associated printer, or Parmsgroup on the LUMAP statement.

Position(5) Set to L or Q indicating that LOGAPPL or QINIT is coded on the LUMAP statement.

Position(6) Set to F if FIRSTONLY coded on the LUMAP statement.

Position(7) Set to D if DEFONLY coded on the LUMAP statement.

Position(8) Set to K if KEEPOPEN coded on the LUMAP statement.

Position(9) Set to C or E indicating Capacity level set or the group is an Exit.

da_name is either the DEFAPPL application name, the associated printer name, or the PARMSGROUP name on the LUMAP statement. Position 4 of the *map_opts* indicates which name is displayed on the line.

persis represents different persistence-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) L if LuSessionPend coded.

Position(2) M if MSG07 coded.

Position(3) T if TkoSpecLu, R if TkoSpecLuRecon coded.

Position(4) Q if QueueSession coded.

Position(5) P if DropAssocPrinter coded.

Position(6) K if KeepLU coded.

function represents different general function options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) O if OldSolicitor coded.

Position(2) A if SingleAttn coded.

Position(3) T if TN3270E coded.

Position(4) S if SnaExt coded.

Position(5) Q if SequentialLU coded.

Position(6) S if SimClientLu coded.

Position(7) W if WlmClusterName coded.

Position(8) H if Hostname lookup will be done.

Position(9) T if TelnetDevice coded.

diag represents different diagnostics-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) T if Debug Trace, D if Debug Detail, S if Debug Summary coded, E if Debug Exception coded.

Position(2) C if Debug routing is to the console, J if Debug routing is to the joblog.

Position(3) F if FullDataTrace coded.

secure represents different security-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) S if SecurePort coded.

Position(2) N if ConnType NegtSecure, B if ConnType Basic, S if ConnType Secure, A if Conntype Any, 0 if Conntype None coded.

Position(3) M if MVS keyring, H if HFS keyring, S if SAF keyring coded.

Position(4) L if Crl LDAP server coded.

Position(5) E if Encryption values coded, D if Default coded.

Position(6) S if Clientauth SSL, F if Clientauth SAF coded.

Position(7) X if ExpressLogon coded.

timers represents different timer-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) I if Inactive timer coded.

Position(2) K if KeepInactive timer coded.

Position(3) P if PrtInactive timer coded.

Position(4) S if ScanInterval timer coded.

Position(5) T if Timemark timer coded.

Position(6) S if SSLtimeout timer coded.

smf represents different SMF options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) I if SMFinit num coded.

Position(2) T if SMFterm num coded.

Position(3) I if SMFinit TYPE119 coded.

Position(4) T if SMFterm TYPE119 coded.

max represents different maximum limit related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) R if MaxReceive coded.

Position(2) S if MaxVtamSendQ coded.

Position(3) Q if MaxReqSess coded.

line represents different Linemode-related options in Telnet. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option was not set or it might be set to asterisk (*), indicating that the particular option was turned off. Otherwise, possible values by position are:

Position(1) B if BinaryLineMode coded.

Position(2) S if SGA coded.

Position(3) C if CodePage tables exist.

Position(4) T if DbcsTransform coded.

Position(5) T if DbcsTrace coded.

source is the source of the parameter line. The details of any of these sources can be seen by issuing a DISPLAY OBJECT, ID=*source*. Possible values are:

***DEFAULT** Represents the default parameter values set by Telnet.

***TGLOBALS** Represents the parameters set in TELNETGLOBALS.

***TPARMS** Represents the parameters set in TELNETPARMS.

TP-profnum Represents the merged results of the DEFAULT, TELNETGLOBALS, and TELNETPARMS parameters. This line is used only on the detailed connection display and matches the profile displays. Parameter details can be seen by issuing a DISPLAY PROF, PROF=*profnum*, DETAIL.

PARMSGROUP names Represents the parameters set in a PARMSGROUP mapped by PARMSMAP to the Client Identifier.

FINAL Represents the merged results of all parms through the PARMSGROUP level, including any PARMSGROUP mapped by the USERID or USERGRP Client Identifier and any parmsgroup mapped on the LUMAP–PMAP combination.

maptype specifies the type of LUMAPping done. The value is either GENERIC or SPECIFIC.

luname is the name of the single LU or LUGROUP on the LUMAP statement that also contains the PMAP option mapping the parmsgroup listed in the next line.

System Action: None

User or Operator Response: None

System Programmer Response: None

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6082I EZZ6082I TELNET CLIENT ID LIST

Message Format:

```
EZZ6082I TELNET CLIENT ID LIST
clid_type
  clid_name             clid_name              clid_name
```

Explanation: EZZ6082I displays a list of Client Identifier names sorted by Client Identifier, profile, and port. The CLIENTID display can be used to see what Client IDs are defined in the profile and some details about the Client ID.

Profile and port divider line information follows each profile/port section. See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details. See message EZZ6081I for an explanation of the summary parameter fields.

clid_type is the type of Client Identifier. Possible client identifier types are:

USERID An exact user ID specified on a mapping statement.

HOSTNAME An exact hostname specified on a mapping statement.

IPADDR An exact IP address specified on a mapping statement.

USERGRP A user ID group name specified on a mapping statement.

HNGRP A hostname group name specified on a mapping statement.

IPGRP An IP address group name specified on a mapping statement.

DESTIP An exact destination IP address specified on a mapping statement.

LINKNAME An exact linkname specified on a mapping statement.

DESTIPGRP A destination IP address group name specified on a mapping statement.

LINKGRP A linkname group name specified on a mapping statement.

NULL The special case where no client identifier is specified.

clid_name is the exact name of the Client Identifier.

System Action: None

User or Operator Response: None

System Programmer Response: None

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6083I EZZ6083I TELNET OBJECT DISPLAY

Message Format:

```
EZZ6083I TELNET OBJECT DISPLAY
OBJECT      CONNS  CLIENT ID CLIENT ID                 ITEM
NAME        USING  TYPE      NAME             OPTIONS  MAPPED
----------  ------ --------- ---------------- -------- --------
obj_type
  obj_name  clconn clid_type clid_name        map_opts da_name
obj_type: obj_name
      obj_entries
```

Explanation: This message is displayed in response to the Telnet Object display command. It displays a detailed list of Object names and their Client Identifiers sorted by Object, profile, and port. The OBJECT display can be used to see what Objects are defined in the profile and some details about the Objects. The Object group details can be displayed when the ID= parameter is used in the display command.

Profile and port divider line information follows each profile/port section. See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist.

obj_type is the type of Object. Possible Object types are:

ARAPPL The Allowappl or Restrictappl application name. If detailed information for one Object name is requested, the second *obj_type* field will indicate that the application is an ALLOWAPPL or a RESTRICTAPPL.

DEFAPPL The application name specified on the DEFAULTAPPL statement.

LINAPPL The application name specified on the LINEMODEAPPL statement.

MAPAPPL The application name specified on the LUMAP statement, DEFAPPL parameter.

PRTAPPL The application name specified on the PRTDEFAULTAPPL statement.

USS The USS table name specified on the USSTCP statement.

INT The interpret table name specified on the INTERPTCP statement.

LU The exact LU name specified on the LUMAP statement.

LUGRP The LUGROUP name specified on the LUMAP statement.

PRT The exact printer LU name specified on the PRTMAP statement.

PRTGRP The PRTGROUP name specified on the PRTMAP statement.

PARMSGRP The PARMSGROUP name specified on the PARMSMAP statement.

obj_name is the exact name of the Object. If Object is LUGROUP or PRTGROUP this line also indicates if the group is an exit or if a capacity warning level was specified.

obj_entries is a list of Objects within a group. For RESTRICTAPPL *obj_types*, the *obj_entries* represent each User ID and optional LU or LUGROUP from the RESTRICTAPPL-USER statement. The LU names follow.

clconn is the number of connections using the Client Identifier mapping statement.

clid_type is the type of Client Identifier. Possible Client Identifier types are:

USERID An exact user ID specified on a mapping statement.

HOSTNAME An exact hostname specified on a mapping statement.

IPADDR An exact IP address specified on a mapping statement.

USERGRP A user ID group name specified on a mapping statement.

HNGRP A hostname group name specified on a mapping statement.

IPGRP An IP address group name specified on a mapping statement.

DESTIP An exact destination IP address specified on a mapping statement

LINKNAME An exact linkname specified on a mapping statement.

DESTIPGRP A destination IP address group name specified on a mapping statement.

LINKGRP A linkname group name specified on a mapping statement.

NULL The special case where no client identifier is specified.

clid_name is the name of the Client Identifier. If the Object is an ALLOWAPPL or a RESTRICTAPPL, the column is used to indicate if DISCONNECTABLE, QSESSION, or CERTAUTH was specified.

DIS Indicates that DISCONNECTABLE was coded on ARAPPL type ALLOWAPPL or RESTRICTAPPL.

QS Indicates that QSESSION was coded on ARAPPL type ALLOWAPPL or RESTRICTAPPL.

CA Indicates that CERTAUTH was coded on ARAPPL type RESTRICTAPPL.

map_opts are additional mapping options on certain mapping statements. Each character position represents a different option. The position might be set to dash (-), indicating that the particular option is not set or does not apply. Possible values by position are:

Position(1) Set to A or R indicating the LUMAP-DEFAPPL is Allowappl or Restrictappl, if LUMAP-DEFAPPL is coded on the LUMAP statement.

Position(2) Set to L if LUs are mapped on the Allowappl or Restrictappl statement that matches the LUMAP-DEFAPPL application name coded on the LUMAP statement.

Position(3) Set to G or S indicating the mapping is Generic or Specific for the LUMAP or PRTMAP statement.

Position(4) Set to D, A, or P indicating that the *da_name* is a Defappl, Associated printer, or Parmsgroup on the LUMAP/PRTMAP statement. Set to M indicating the object is specified on the LUMAP/PRTMAP statement. In this case, the item mapped column displays the LU or group object on the mapping statement.

Position(5) Set to L or Q indicating that LOGAPPL or QINIT is coded on the LUMAP statement.

Position(6) Set to F if FIRSTONLY is coded on the LUMAP statement.

Position(7) Set to D if DEFONLY is coded on the LUMAP statement.

Position(8) Set to K if KEEPOPEN is coded on the LUMAP statement.

Position(9) Set to C or E indicating Capacity level set or the group is an Exit.

da_name is either the DEFAPPL application name, the associated printer name, or the PARMSGROUP name on the LUMAP statement. Position 4 of the map_opts indicates which name is displayed on the line.

System Action: None

User or Operator Response: None

System Programmer Response: None

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6084I EZZ6084I TELNET OBJECT LIST

Message Format:

```
EZZ6084I TELNET OBJECT LIST
obj_type
  obj_name  obj_name  obj_name  obj_name  obj_name
```

Explanation: EZZ6084I displays a list of Object names sorted by Object type, profile, and port. The OBJECT display can be used to see what Objects are defined in the profile and some details about the Oblects.

Profile and port divider line information follows each profile/port section. See message EZZ6050I for details. The last line of this message indicates how many lines of data were displayed and how many total lines exist. See EZZ6052I for details. See message EZZ6081I for an explanation of the summary parameter fields.

obj_type is the type of Object. Possible Object types are:

ARAPPL The Allowappl or Restrictappl application name.

DEFAPPL The application name specified on the DEFAULTAPPL statement.

LINAPPL The application name specified on the LINEMODEAPPL statement.

MAPAPPL The application name specified on the LUMAP statement, DEFAPPL parameter.

PRTAPPL The application name specified on the PRTDEFAULTAPPL statement.

USS The USS table name specified on the USSTCP statement.

INT The interpret table name specified on the INTERPTCP statement.

LU The exact LU name specified on the LUMAP statement.

LUGRP The LUGROUP name specified on the LUMAP statement.

PRT The exact printer LU name specified on the PRTMAP statement.

PRTGRP The PRTGROUP name specified on the PRTMAP statement.

PARMSGRP The PARMSGROUP name specified on the PARMSMAP statement.

obj_name is the exact name of the Object.

System Action: None

User or Operator Response: None

System Programmer Response: None

Source Data Set: EZAZMTNS

Procedure Name: TCP/IP

EZZ6201I SNMP agent: Unable to open message catalog *snmpdmsg.cat***:** *additional error text*

Explanation: The SNMP agent was unable to open the message catalog *snmpdmsg.cat* in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be *NLSPATH=/usr/lib/nls/msg/%L/%N*.

System Action: The agent will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *z/OS C/C++ Run-Time Library Reference* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the *z/OS UNIX System Services Programming Tools*. If the default messages are acceptable, no action is necessary.

Source Data Set: EZASNAA3.C, EZASNPWT.C

Procedure Name: main

EZZ6202I Using catalog *catalog file* **for SNMP agent messages**

Explanation: The SNMP agent located its message catalog file.

System Action: The agent continues to initialize.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6203I Sigaction for *signal handler* **failed :** *specified action: error text (errno/errnojr)*

Explanation: The agent issued a sigaction function for the *specified action* which failed.

error text provides more information about the cause of the error.

errno is the decimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

signal handler is one of the following sigaction functions: SIGTTOU, SIGTERM, SIGABEND, SIGPIPE, SIGTTIN.

System Action: The agent continues to initialize.

User or Operator Response: None.

System Programmer Response: Correct the problem indicated by *error*. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6204I SIGTERM received for SNMP daemon which is now shutting down.

Explanation: The agent received a SIGTERM signal.

System Action: The agent ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: pgmstpd

EZZ6205I SNMP agent: Could not determine TCPIP jobname using default of 'INET'

Explanation: The SNMP agent, osnmpd, could not determine the jobname for the TCP/IP stack that it is to associate itself with. A default value of 'INET' will be used for TCPIPjobname.

System Action: SNMP agent continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for the SNMP agent to communicate with a particular stack, the TCPIPjobname should be set in the appropriate resolver configuration file or data set. See the *z/OS Communications Server: IP Configuration Guide* for information about file search order.

An SNMP agent must be associated with a single TCP/IP instance, since a portion of the MIB objects supported by a TCP/IP instance are actually implemented in the agent.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6206I Unable to open *configuration file* **:** *error text*

Explanation: The agent was unable to open the *configuration file*. This *configuration file* can be the community names file or the file for the default MIB variables. Refer to the *error text* for a more specific reason for the failure. The community names file or the MIB variables file can be in one of several files. The initialization process is searching for a community names or MIB variables file in the prescribed order and reporting on its status.

System Action: The agent continues searching for the next file name.

User or Operator Response: This is an informational message indicating that the particular file in the path was not found.

System Programmer Response: A community names or MIB variables file is required for changing defaults. If the indicted file was not found at all, then the configuration of the agent ends and the agent continues with initialization. Refer to the *z/OS Communications Server: IP Configuration Guide* for configuring the community names data set.

Source Data Set: EZASNAC3.C, EZASNASO.C, EZASNAA3.C

Procedure Name: configure_v1_default, configure_community, set_mib_install_defaults

---

EZZ6207I Line number *number* in the *community name* file is not in the correct format

Explanation: The agent was unable to interpret the statement in the *community name* file. The statement had fewer than three fields, (community name, IP address, and mask) or was not a comment statement that started with a '*' or '#' character.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None..

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. If necessary, reIPL the agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information.

Source Data Set: EZASNASO.C

Procedure Name: configure_community

EZZ6208I Line number *number* in the *community name* file has a community name greater than *number* characters

Explanation: The community name at the specified line number was greater than the indicated number of characters. The statement is ignored and configuration continues.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. In order to pick up the changes, reIPL the agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the community names data set.

Source Data Set: EZASNASO.C

Procedure Name: configure_community

EZZ6209I Line number *number* in the *community name* file has a network address: *IP address* which is not in the correct format

Explanation: The network address at the specified line number is not in dotted notation(xxx.xxx.xxx.xxx). The statement is ignored and configuration continues.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the community names data set.

Source Data Set: EZASNASO.C

Procedure Name: configure_community, configure_COMMUNITY_entry

EZZ6210I Line number *number* in the *community name* file has a network mask: *IP address* which is not in the correct format

Explanation: The network mask at the specified line number is not in dotted notation(xxx.xxx.xxx.xxx). The statement is ignored and configuration continues.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the configuration and correct it. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the community names data set.

Source Data Set: EZASNASO.C

Procedure Name: configure_community, configure_COMMUNITY_entry

EZZ6211I Error reading the *community names* **file at line number:** *number* **error text:** *error text*

Explanation: The agent was attempting to read the specified line number in the *community names* file but had an error described by the *error text*. Refer to the *error text* for a more specific reason for the failure. The error number is also supplied.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: Determine why there was an i/o error and correct it.

Source Data Set: EZASNAA3.C, EZASNASO.C

Procedure Name: set_mib_install_defaults, configure_community

EZZ6212I Unable to open *trap destination* **file :** *error text*

Explanation: The agent was unable to open the *trap destination* file. Refer to the *error text* for a more specific reason for the failure. The trap destination file can be in one of several files. The initialization process is searching for a trap names file in the prescribed order and reporting on its progress attempting to open the files.

System Action: The agent continues searching for the next trap destination file.

User or Operator Response: None.

System Programmer Response: This message can be ignored as long as a trap destination file is eventually found. If no trap destination file is found, then defaults will be used and message EZZ6213I will be issued for this condition.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6213I Using SNMP trap defaults

Explanation: The agent was unable to open any trap destination files.

System Action: The agent uses the default trap destination settings.

User or Operator Response: None.

System Programmer Response: The agent continues initializing. Refer to the *z/OS Communications Server: IP Configuration Guide* for creating a trap destination data set.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6214I Using *trap destination file* **for trap destination file**

Explanation: The agent found and opened the *trap destination file*.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6215I Line number *number* **in the** *trap destination* **file is not in the correct format**

Explanation: Trap initialization could not process the statement identified by the indicated line number.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the *trap destination* file and correct

it. Restart the agent after the statement was corrected. Refer to the *z/OS Communications Server: IP Configuration Reference* on creating a trap destination data set.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6216I line number *number* **in the** *trap destination* **file failed host_lookup for ip address:** *hostname*

Explanation: The host_lookup function could not resolve the *hostname* into an ip address for the given *trap destination* at line number *number* in the *trap destination* file.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement in the *trap destination* file and correct it. Restart the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6217I Line number *number* **in the** *trap destination* **file** *protocol* **is not a supported connection**

Explanation: The only trap protocol for trap reporting is **UDP**. At indicated line number in the trap destination file something other than **UDP** was specified.

System Action: The agent ignores the current statement and continues reading in the next statement in the file.

User or Operator Response: None.

System Programmer Response: Determine what is wrong with the statement at line number *number* in the *trap destination* file and correct it.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6218I Error reading the *trap destination* **file at line number:** *line number* **- error text:** *error text*

Explanation: There was an error reading the *trap destination* file at the line indicated by *line number*. The *error text* will provide more information about the cause of the error.

System Action: The agent closes the *trap destination* file current statement and continues initialization.

User or Operator Response: None.

System Programmer Response: Refer to the error text to determine what is wrong with the *trap destination* file.

Source Data Set: EZASNASO.C

Procedure Name: configure_trap_v1

EZZ6219I rc=*rc* **(***error text***) (expect** *rc* **(***additional error text***)) from (***SNMP function name***)**

Explanation: There was an error while the agent attempted to add a target address entry for a notification destination.

SNMP function name is the failing SNMP function name.

error text and *type* provide more information about the cause of the error.

System Action: The agent ignores the problem and continues with initialization.

User or Operator Response: None.

System Programmer Response: Try to correct the problem with the help of the information in this message.

Source Data Set: EZASNAC3.C

Procedure Name: add_notification_destination

EZZ6220I Unknown notification version *version*

Explanation: There was an error while the agent attempted to add a notification destination. The *version* was specified and it is not recognized as valid.

System Action: The agent ignores the problem and continues with initialization.

User or Operator Response: None.

System Programmer Response: Try to correct the problem with the help of the information in this message. See the *z/OS Communications Server: IP Configuration Reference* for SNMP agent trap destination configuration.

Source Data Set: EZASNAC3.C

Procedure Name: add_notification_destination

EZZ6221I Need community name

Explanation: When the agent was started a **-c** argument was specified but there was no community name specified.

System Action: The agent ended.

User or Operator Response: None.

System Programmer Response: A community name must be specified if the **-c** argument is specified. The community name follows the **-c** argument. For more information, see the *z/OS Communications Server: IP Configuration Reference*

Source Data Set: EZASNAC3.C

Procedure Name: snmp_config_parse_options

EZZ6222I Need a UNIX socket name

Explanation: When the agent was started a **-s** argument was specified but there was no UNIX socket name specified.

System Action: The agent ended.

User or Operator Response: None.

System Programmer Response: A UNIX socket name must be specified if the -s argument is specified. The UNIX socket name follows the **-s** argument.

Source Data Set: EZASNAC3.C

Procedure Name: snmp_config_parse_options

EZZ6225I SNMP agent: Initialization complete

Explanation: The SNMP agent completed initialization and is ready to receive requests.

System Action: The agent is functioning.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6226I *port number* **is an incorrect port number, using** *default port*

Explanation: When the agent was started, a **-p** argument specified a *port number* which was not valid port. The default port value, *default port* is used instead.

System Action: The agent continues initializing.

User or Operator Response: The user can stop the agent and restart it with the intended port number.

System Programmer Response: None.

Source Data Set: EZASNAC3.C

Procedure Name: snmp_config_parse_options

EZZ6227I Added managers for community: *community name* **rc=** *rc*

Explanation: When the agent attempted to add the community name, it found that the name already existed. As a result, it becomes a multiple manager community name.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAC3.C, EZASNASO.C

Procedure Name: configure_v1_default, configure_community, configure_COMMUNITY_entry

---

EZZ6228I (*error text***) from snmp_local_config_set()**

Explanation: The agent was attempting to set the DPI MIB variable, dpiPortForTCP, but was not successful. The *error text* will provide more information about the cause of the error. All sub-agents that use this connection will be unable to communicate with the agent. This does not affect UNIX stream subagent connections.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: set_dpiport

EZZ6229I Closing DPI UNIX socket connection, fd=*socket file descriptor*

Explanation: The agent closed a DPI UNIX socket connection to a sub-agent on the file descriptor *socket file descriptor*. This is normally a message that reports agent activity.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: close_dpi

EZZ6230I Closing DPI inet socket connection, fd=*socket file descriptor***, address** *ip address* **port** *port number*

Explanation: The agent closed a DPI inet socket connection to a sub-agent on the file descriptor *socket file descriptor*. *ip address* is the ip address and *port number* is the port number. This is normally a message that reports agent activity.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: close_dpi

EZZ6231I (*error text*) from snmp_delete_subagent()

Explanation: The agent closed a DPI socket connection and then tried to remove all control blocks related to this sub-agent but had a problem. The *rc* was returned from the snmp_delete_subagent function. The particular problem is specified by the additional *error text* error text.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: close_dpi

EZZ6232I The SNMP agent is running as jobname, *jobname*.

Explanation: The agent is reporting its jobname.

System Action: The agent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6233I Cannot allocate buffers for SNMP and DPI packets; SNMP agent ended.

Explanation: The agent cannot allocate enough space for buffers.

System Action: The agent ends.

User or Operator Response: None.

System Programmer Response: Determine why there is not enough main memory.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6234I gethostname function failed; SNMP agent defaulting to 127.0.0.1

Explanation: The gethostname function failed. The agent will default to using the loopback address as its local address.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: Determine why the system cannot get its hostname if use of the loopback address is not acceptable.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6235I *function name and socket type failure*; *error text*

Explanation: The failing *function name and socket type* The *error text* will provide more information about the cause of the error.

System Action: The agent may end if it involves SNMP socket(161).

User or Operator Response: None.

System Programmer Response: Determine why the system function failed.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6236I bind function failed for SNMP inet udp socket; *error text*

Explanation: The bind function failed to get a socket that would be used for communication between the agent and manager functions. The *error text* will provide more information about the cause of the error.

System Action: The agent ends.

User or Operator Response: None.

System Programmer Response: Determine why the system cannot do a bind.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6237I *failing_socket_function_name for a socket_type*; *error text*

Explanation: The socket function failed for a socket that would be used for communication between the agent and sub-agents functions. The *error text* will provide more information about the cause of the error.

System Action: The agent may end depending on the failing function.

User or Operator Response: None.

System Programmer Response: Determine why the function failed.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6238I *socket function failed for UNIX DPI socket*; *error text*

Explanation: The *socket function failed for UNIX DPI socket* which is used for communication between the agent and sub-agents functions. The *error text* will provide more information about the cause of the error.

The SNMP agent is attempting to establish an AF_UNIX socket for use in commuicating with subagents. The AF_UNIX socket is represented by a file name, which defaults to /tmp/dpi_socket. The file name may be specified either on the **-s** parameter at SNMP agent initialization or by setting the DPIPATHNAMEFORUNIXSTREAM value in the OSNMPD.DATA file.

System Action: The agent may end depending on the socket function.

User or Operator Response: Restart the SNMP agent with tracing by specifying **-d** 255 at agent invocation. Use the errnojr shown in the traces for additional help in diagnosing the problem.

System Programmer Response: Determine why the socket function is failing.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6239I selectex() function timed out on wait forever

Explanation: The selectex socket call made by the SNMP agent timed out. However, the call made by the SNMP agent requested that it wait forever, so the selectex should never have timed out. It might indicate a problem with the SNMP agent, the selectex socket call or the TCP/IP stack.

System Action: The SNMP agent tolerates a small, fixed number of errors on the selectex call before ending. If the count of selectex errors is within the limit, the SNMP agent reissues the selectex call. If the error limit was exceeded, the SNMP agent ends.

User or Operator Response: This condition should be reported to your system administrator. collect traces for the SNMP agent, the C socket layer, and the TCP/IP stack to diagnose the error. Restart the SNMP agent with the traces enabled.

System Programmer Response: Collect the traces and contact the IBM Software Support Center.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6240I selectex() function failed return code = *rc*

Explanation: The selectex socket call made by the SNMP agent failed with the return code shown. This might indicate a problem with the SNMP agent, the selectex socket call, or the TCP/IP stack. For example, if the TCP/IP stack is shut down before the SNMP agent is stopped, this message will be issued.

rc is the return code for the selectex socket call. These return codes are documented in the *z/OS C/C++ Run-Time Library Reference*.

System Action: The SNMP agent tolerates a small, fixed number of errors on the selectex call before ending. If the count of selectex errors is within the limit, the SNMP agent reissues the selectex call. If the error limit was exceeded, the SNMP agent ends.

User or Operator Response: This condition should be reported to your system administrator.

If this error is not caused by the TCP/IP stack being shut down, then report this condition to your system administrator. Restart the SNMP agent with traces for the SNMP agent, the C socket layer, and the TCP/IP stack enabled to diagnose the error.

System Programmer Response: Contact the IBM Support Center and supply the traces collected.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6241I receive function failed; *error text*

Explanation: The receive function failed. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6242I accept function failed for a DPI inet socket; *error text*

Explanation: The accept function failed for a DPI inet socket that would be used for communication between the agent and sub-agent functions. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6243I Refused new DPI connection, at maximum connection of *maximum_agents*

Explanation: The agent refused the connection of a new sub-agent because of a limit on the number of active sub-agents.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6244I Accepted new DPI inet socket connection on fd=*file descriptor* **from** *inet address* **port** *port number*

Explanation: The agent accepted a connection to a sub-agent. This event is logged.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6245I accept function failed for a DPI UNIX socket; *error text*

Explanation: The accept function failed for a DPI UNIX socket that would be used for communication between the agent and sub-agent functions. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6246I Accepted new DPI UNIX socket connection on fd=*file descriptor***.**

Explanation: The agent accepted a connection to a sub-agent over a UNIX socket connection. This event is logged.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6247I SNMP agent terminating, got *number* **errors on selectex**

Explanation: The selectex function failure count reached its limit.

System Action: The agent ends.

User or Operator Response: None.

System Programmer Response: Determine why the system is getting selectex failures.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6248I A select function timed out on a DPI connection

Explanation: The receive function failed. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: mustread

EZZ6249I EOF received on DPI fd *number*

Explanation: The receive function returned an EOF for the indicated file descriptor *number* while trying to receive from a subagent.

System Action: The DPI connection is closed. The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: read_tcp_dpi_packet

EZZ6250I receive function failed for a DPI socket; *error text*

Explanation: The receive function failed for a DPI socket. The *error text* indicates the specific reason for failure.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: read_tcp_dpi_packet

EZZ6251I sendto function failed for a DPI socket; *error text***.**

Explanation: The sendto function failed for a DPI socket. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: send_snmp_message

EZZ6252I On a sendto function only *number sent* **out of** *total number* **bytes sent.**

Explanation: The sendto function failed to send all the intended bytes on a DPI socket.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: asend_snmp_message

EZZ6253I send function failed; *error text*

Explanation: The send function failed on a DPI socket. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: send_message_to_snmp_subagent

EZZ6254I On a send function only *number* **out of** *number* **bytes sent**

Explanation: The send function failed to send all the intended bytes on a DPI socket.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: send_message_to_snmp_subagent

EZZ6255I select function failed return code = *rc*

Explanation: The select socket call made by the SNMP agent failed with the return code shown. This might indicate a problem with the SNMP agent, the select socket call, or the TCP/IP stack.

rc is the return code for the select socket call. These return codes are documented in the *z/OS C/C++ Run-Time Library Reference*.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: await_message_from_snmp_subagent

EZZ6256I gethostbyname function failed for *hostname***;** *error text*

Explanation: The gethostbyname function failed. The *error text* will provide more information about the cause of the error.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: Determine if *hostname* is a valid hostname.

Source Data Set: EZASNAA3.C

Procedure Name: snmp_IPaddress LINKAGE lookup_host

EZZ6257I SIGPIPE received for an SNMP agent which is now shutting down

Explanation: A SIGPIPE signal was received by the agent.

System Action: The agent ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: sigpipedd

EZZ6258I SIGABND signal received. SNMP agent daemon terminating with *abendcode***,** *rsncode***.**

Explanation: A SIGABND signal was received by the agent.

abendcode is the hexadecimal MVS Abend code. MVS abend codes are described in the *z/OS MVS System Codes*.

reasoncode is the reason code associated with this *abendcode*. They are listed in the description of the abend code in the *z/OS MVS System Codes*.

System Action: The agent ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: pgmabndd

EZZ6259I Tracing is set to *trace setting*

Explanation: This a response to a modify command that indicates the current trace setting.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: mvs_command_handler

EZZ6260I Unrecognized modify request

Explanation: This a response to a modify command that could not be serviced because the request was not recognized.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: mvs_command_handler

EZZ6261I Modify request completed

Explanation: This a response to a modify command that completed.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: mvs_command_handler

EZZ6262I Unsupported modify command received

Explanation: The agent modify command handler does not support the requested modify.

System Action: The agent continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: mvs_command_handler

EZZ6263I Configuration problem in file *filename* **at line** *line number* **variable name** *MIB variable name* **not recognized.**

Explanation: The agent was attempting to configure the MIB variables from the configuration file and did not recognize the MIB variable at the indicated line number.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: Check the file and correct the MIB variable at that line number.

Source Data Set: EZASNAA3.C

Procedure Name: set_mib_install_defaults

EZZ6264I Configuration problem in file *file name* **at line** *line number* **variable name** *variable name* **the value:** *value* **not recognized.**

Explanation: The agent was attempting to configure the variable from the configuration file and did not recognize the variable at the indicated line number.

System Action: The agent continues initializing.

User or Operator Response: None.

System Programmer Response: Check the file and correct the MIB value at that line number.

Source Data Set: EZASNAA3.C

Procedure Name: set_mib_install_defaults

EZZ6265I Usage: *command* **[options], incorrect option specified. To see a complete list of options, enter** *command* **?**

Explanation: This is a list of all supported options.

In the message text, the *options* are:

-a send packets using physical interface address as source address

-c *community*
a community name or password for SNMP requests (default public)

-d *level* debug level [0...255] default 0 (but 31 is used if **-d** is specified without a value)

-p *port* listen for SNMP packets on this port (default 161)

-? display the usage statement

System Action: Processing ends.

User or Operator Response: Correct the option or version and reissue the command.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: usage

EZZ6266I Using file *file name* **for community names configuration**

Explanation: The agent is using the *file name* as its source of community name configuration.

System Action: None.

User or Operator Response: None.

System Programmer Response: The agent configures the community name from entries in this file.

Source Data Set: EZASNASO

Procedure Name: configure_community

EZZ6267I Tracing set to *level*

Explanation: This message indicates what tracing level was requested. There may be several following lines in the syslog that further state each type of tracing requested.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAC3.C

Procedure Name: snmp_config_parse_options

EZZ6268I pwtokey usage statement

Explanation: The pwtokey usage statement is displayed when pwtokey is invoked with the **-?** option.

System Action: pwtokey ends after displaying the usage statement.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNPWT.C

Procedure Name: do_usage

EZZ6269I Display of *number* **byte** *description***:**

Explanation: The pwtokey function is writing out a value (either key or SNMP EngineID) as described by the textual description. The value is displayed on the following line.

System Action: pwtokey function continues.

User or Operator Response: The output of the pwtokey function is used in configuring the SNMP agent for SNMPv3 security.

System Programmer Response: None.

Source Data Set: EZASNPWT.C

Procedure Name: dump_bfr

EZZ6272I SNMP agent: Could not establish affinity with '*jobname***' (***errno***/***errnojr***)**

Explanation: The SNMP agent cannot communicate with the TCP/IP stack *jobname*. The SNMP agent attempted to use the socket call, setibmopt(), to associate itself with the TCP/IP instance *tcpip_name*. This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifer' format of the MVS Start command was used) of the TCP/IP instance with which the SNMP agent is to be associated. The setibmopt call failed with the displayed *errno* and *errnojr*.

errno is the decimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The SNMP agent ends abnormally.

User or Operator Response: Most likely, the TCP/IP instance's name was not defined correctly to OMVS. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure OMVS. Ensure that the TCP/IP started procedure name (or identifier if the 'S member.identifier' format of the MVS Start command was used) matches the SUBFILESYSTYPE NAME. Recycle OMVS or TCP/IP if a change is necessary. If none of the above error conditions exist contact your system programmer.

System Programmer Response: For the SNMP agent to communicate with a particular stack, the jobname (as determined by the system variable TCPIPjobname) must match ″xxxxx″ where ″xxxxx″ is set in the BPXPRMPx member used to start OMVS. ″xxxxx″ is set in the SUBFILESYSTYPE NAME(xxxxx) for ENTRYPOINT(EZBPFINI). In order to establish an affinity with a corresponding TCPIP stack, the SNMP agent uses the setibmopt call. Correct the error indicated by *error_code* and *reason*.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6273I Value specified for the *startup* parameter is missing

Explanation: When the agent was started a '-' argument was specified without a value. The default value continues to be in effect.

System Action: The agent continues initializing.

User or Operator Response: The user can stop the agent and restart it with the intended value.

System Programmer Response: None.

Source Data Set: EZASNAC3.C

Procedure Name: snmp_config_parse_options

EZZ6275I SNMP agent: using file *file name* for configuration

Explanation: This SNMP agent is using *file name* as its source for user, view, access list, community name, and trap destination definitions.

System Action: The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAC3.C

Procedure Name: snmp_configure

EZZ6276I SNMP agent: using file *file name* for MIB variable defaults

Explanation: This SNMP agent is using *file name* as its source for default MIB variable values.

System Action: The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNAA3.C

Procedure Name: set_mib_install_defaults

EZZ6277I SNMP agent: unable to open or create boots file *file name* : *error text*

Explanation: This SNMP agent is attempting to initialize security settings using *file name*. If the file does not exist, it will be created. If it cannot be created, this message is issued. Refer to the *error text* for a more specific reason for the failure.

System Action: The SNMP agent will continue to attempt to open all files in the search path for the boots initialization file.

User or Operator Response: None.

System Programmer Response: A boots initialization file is required when SNMPv3 security is used. This is an informational message indicating which file was attempted in the search order. Refer to the *z/OS Communications Server: IP Configuration Guide* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: configure_engine_boots

EZZ6278I SNMP agent: unsuccessful initialization: *(no boots file | no engineID | incorrect engineID | incorrect engineBoots)***. SNMP agent ending.**

Explanation: This SNMP agent is attempting to initialize for SNMPv3 security, but one of the following occurred:
- no boots file - no boots file could be opened or created.
- no engineID - an existing boots file was opened, but it did not contain an engineID value.

- incorrect engineID - an existing boots file was opened, but it contained an engineID that was not in the correct format.
- incorrect engineBoots - an existing boots file was opened, but it contained an engineBoots value that was not in the correct format.

System Action: The SNMP agent ends.

User or Operator Response: None.

System Programmer Response: A boots initialization file is required when SNMPv3 security is used. If an existing file is opened, it must contain valid engineID and engineBoots values. Either correct the error and restart the agent or configure the SNMP agent not to use SNMPv3 security. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: configure_engine_boots

EZZ6279I SNMP agent: using *(new | existing)* **file** *file name* **for boots initialization**

Explanation: The file used for SNMP agent boots initialization is displayed with an indication of whether the agent created the file or used an existing file.

System Action: The SNMP agent continues boots initialization.

User or Operator Response: None.

System Programmer Response: To ensure security of SNMPv3 requests, it is recommended that the same file be used for boots initialization across restarts of the SNMP agent. This message indicates whether a new file was created or an existing file was used.

Source Data Set: EZASNASO.C

Procedure Name: configure_engine_boots

EZZ6280I SNMP agent: engineBoots value is wrapping

Explanation: The SNMP engineBoots value read from the boots file was greater than or equal to the maximum possible engineBoots value (2,147,483,647). It is incremented each time the agent restarts.

System Action: The SNMP agent terminates.

User or Operator Response: None.

System Programmer Response: To ensure security of SNMPv3 requests, the engineBoots value is incremented on each restart of the agent. When the engineBoots value reaches the maximum, good security practices suggest that all configured user secrets (keys) be regenerated before restarting the SNMP agent. Erase the boots file or change the value of engineBoots in the file to zero. Then restart the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: configure_engine_boots

EZZ6281I SNMP agent: line number *number* **contains an unsupported keyword** *keyword*

Explanation: The SNMP agent configuration file contained an unsupported keyword.

System Action: The entry is ignored, and SNMP agent initialization continues with the next line in the file.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: configure_defined_entries

EZZ6282I SNMP agent: line number *number* **contains an unsupported value** *value*

Explanation: The SNMP agent configuration file contained an unsupported value on the line indicated.

System Action: The entry is ignored, and SNMP agent initialization continues with the next line in the file.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: process_DEFAULT_SECURITY_entry, configure_USM_USER_entry, configure_VACM_VIEW_entry, configure_VACM_ACCESS_entry, configure_NOTIFY_entry, configure_TARGET_ADDRESS_entry, configure_TARGET_PARAMETERS_entry, configure_COMMUNITY_entry, validate_securityModel_on_entry, validate_securityLevel_on_entry, validate_storageType_on_entry, configure_NOTIFY_FILTER_entry

EZZ6283I SNMP agent: line number *number* **contains an incorrect number of keywords for a** *entry type* **entry**

Explanation: The SNMP agent configuration file requires all keywords to be specified or defaulted with a dash (**-**). The indicated line contains an incorrect number of keywords for the type of entry.

System Action: The entry is ignored. The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: process_DEFAULT_SECURITY_entry, configure_USM_USER_entry, configure_VACM_GROUP_entry, configure_VACM_VIEW_entry, configure_VACM_ACCESS_entry, configure_NOTIFY_entry, configure_TARGET_ADDRESS_entry, configure_TARGET_PARAMETERS_entry, configure_COMMUNITY_entry, configure_NOTIFY_FILTER_PROF_entry, configure_NOTIFY_FILTER_entry

EZZ6284I SNMP agent: line number *number* **contains a keyword value that is too long:** *keyword*

Explanation: The indicated keyword in the SNMP agent configuration file is too long.

System Action: The entry is ignored. The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: process_DEFAULT_SECURITY_entry, configure_USM_USER_entry, configure_VACM_GROUP_entry, configure_VACM_VIEW_entry, configure_VACM_ACCESS_entry, configure_NOTIFY_entry, configure_TARGET_ADDRESS_entry, configure_TARGET_PARAMETERS_entry, configure_COMMUNITY_entry, configure_NOTIFY_FILTER_PROF_entry, configure_NOTIFY_FILTER_entry

EZZ6285I SNMP agent: line number *number* **defaults positional keyword** *keyword* **but that keyword is required.**

Explanation: The SNMP agent configuration file contained a dash (**-**) to default a keyword for which an explicit value must be specified. In some cases, this message will be issued because the value of a keyword makes another keyword required. For example, specifying or defaulting the authProto value to 'HMAC-MD5' makes a key required for authKey.

System Action: The entry is ignored. The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the

z/OS Communications Server: IP Configuration Reference for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: process_DEFAULT_SECURITY_entry, configure_USM_USER_entry, configure_VACM_GROUP_entry, configure_VACM_VIEW_entry, configure_VACM_ACCESS_entry, configure_NOTIFY_entry, configure_TARGET_ADDRESS_entry, configure_TARGET_PARAMETERS_entry, configure_COMMUNITY_entry, configure_NOTIFY_FILTER_PROF_entry, configure_NOTIFY_FILTER_entry

---

EZZ6286I SNMP agent: line number *number* **contains an unsupported value** *value***. Default value is used.**

Explanation: The SNMP agent configuration file contained an unsupported value on the line indicated. The default value was applied and processing of the entry continues.

System Action: The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNASO.C

Procedure Name: configure_TARGET_ADDRESS_entry

EZZ6287I SNMP agent: error configuring *keyword* **entry on line number** *number***. SNMP agent ending.**

Explanation: An error occurred in trying to create a configuration definition for the specified type of entry on the indicated line of the SNMP agent configuration file. The most likely cause of this error is that there was not sufficient memory to create the entry.

System Action: The SNMP agent ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNASO.C

Procedure Name: configure_USM_USER_entry, configure_VACM_GROUP_entry, configure_VACM_VIEW_entry, configure_VACM_ACCESS_entry, configure_NOTIFY_entry, configure_TARGET_ADDRESS_entry, configure_TARGET_PARAMETERS_entry, configure_NOTIFY_FILTER_PROF_entry, configure_NOTIFY_FILTER_entry

---

EZZ6289I SNMP agent: line number *number* **defines a duplicate entry.**

Explanation: A duplicate configuration definition for the SNMP agent was found on the specified line number. The SNMP agent continues initialization.

System Action: The SNMP agent accepts the definition and continues initialization.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNASO.C

Procedure Name: configure_USM_USER_entry, configure_VACM_GROUP_entry, configure_VACM_VIEW_entry, configure_VACM_ACCESS_entry, configure_NOTIFY_entry, configure_TARGET_ADDRESS_entry, configure_TARGET_PARAMETERS_entry, configure_NOTIFY_FILTER_PROF_entry, configure_NOTIFY_FILTER_entry

---

EZZ6290I setibmsockopt function failed for SNMP INET UDP socket; *error text*

Explanation: The setibmsockopt function failed for the SNMP INET UDP socket; The *error text* will provide more information about the cause of the error.

System Action: The agent continues, but SOURCEVIPA will not be ignored on the socket used by the agent to receive and respond to requests. As a result, SNMP requests may fail due to IP address verification that would otherwise have been accepted.

User or Operator Response: None.

System Programmer Response: Determine why the system function failed. Correct the problem and restart the SNMP agent.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6291I SNMP agent initialization terminated because TCP/IP Base feature of OS/390 is not enabled.

Explanation: The SNMP agent issued a query for product=OS/390, feature=TCP/IP_BASE but IFAPRDxx parmlib member does not indicate that the feature TCP/IP_BASE is enabled. This SNMP agent will only run if TCP/IP_BASE is enabled.

System Action: The SNMP agent initialization is terminated.

User or Operator Response: Contact the system programmer.

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

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6292I SNMP agent: unexpected error adding *entry* **entry at initialization. Return code =** *code*

Explanation: The SNMP agent was attempting to add a configuration entry of the specified type at agent initialization when an unexpected error occurred. The error return code is shown.

System Action: The SNMP agent initialization continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Attempt to correct the problem and restart the SNMP agent.

Source Data Set: EZASNAC3.C

Procedure Name: add_notification_destination

EZZ6293I SNMP agent: line number *number* **contains a keyword value that is too short:** *keyword*

Explanation: The indicated keyword in the SNMP agent configuration file is too short.

System Action: The entry is ignored. The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: process_DEFAULT_SECURITY_entry, configure_USM_USER_entry

---

EZZ6294I SNMP agent: line number *number* **defines a** *keyword* **that is not consistent with the** *other keyword* **specification.**

Explanation: A configuration definition on the specified line number contains keyword values that conflict. Possible cases are:

- A USM_USER configuration definition contains a keyType definition that is not consistent with the engineID specification. For example, if the asterisk is used to specify 'any' for the engineID, the keyType must be non-localized.
- A USM_USER configuration definition contains a storageType definition that is not consistent with the authProto specification. For example, the storageType of readOnly is not allowed when an authProto other than 'none' is specified. keyType must be non-localized.

System Action: The SNMP agent ignores the entry and continues initialization.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZASNASO.C

Procedure Name: configure_USM_USER_entry

EZZ6295I SNMP agent: Dynamic configuration initialized.

Explanation: Dynamic configuration for the SNMP agent successfully initialized.

System Action: The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: S@DYNINI.C

Procedure Name: dynInitialize

EZZ6296I SNMP agent: Error in allocating memory in *function*

Explanation: The SNMP agent attempted to allocate memory while running in function *function*. Memory could not be obtained.

System Action: If the error occurs during SNMP agent initialization, the SNMP agent will terminate. If the error occurs while the SNMP agent is processing a SET request, the SNMP agent fails the SET request.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why memory was not available. Correct the problem and restart the SNMP agent.

Source Data Set: S@DYNINI.C, S@DYNCFY, S@DYNFUP

Procedure Name: dynInitialize, dynInitializeConfigList, dynUpdateConfig, dynBuildUsmUser, dynBuildVacmGroup, dynBuildVacmAccess, dynBuildNotify dynBuildTargetAddr, dynBuildTargetParm, dynBuildTxtUserRec, dynBuildTxtGroupRec, dynBuildTxtViewRec, dynBuildTxtAccessRec, dynBuildTxtNotifyRec, dynBuildTxtTargetAddrRec, dynBuildTxtTargetParmRec, dynBuildNotifyFilterProf, dynBuildNotifyFilter, dynBuildTxtFilterProfRec, dynBuildTxtFilterRec

---

EZZ6297I SNMP agent: Error adding entry to dynamic configuration list.

Explanation: The SNMP agent was attempting to add an entry to the dynamic configuration list representing the entries of the SNMPD.CONF configuration file. See previous error messages for the details of the error that occurred.

System Action: If the error occurs during SNMP agent initialization, the SNMP agent will terminate. If the error occurs while the SNMP agent is processing a SET request, the SNMP agent fails the SET request.

User or Operator Response: Contact the system programmer.

System Programmer Response: Depending on the error, either restart the SNMP agent or attempt the SET request again with valid data.

Source Data Set: S@DYNINI.C

Procedure Name: dynInitializeConfigList

EZZ6298I SNMP agent: Unknown record types in *function*

Explanation: The SNMP agent dynamic configuration processing encountered an unrecognized record type in function *function.*

System Action: Collect SNMP agent traces by restarting the SNMP agent with the **-d** prarmeter set to 255. Provide the traces and the SNMP agent configuration file, SNMPD.CONF, to the system programmer.

User or Operator Response: Contact the IBM Software Support Center with the traces and the SNMPD.CONF file.

System Programmer Response: Contact the IBM Support Center for assistance.

Source Data Set: S@DYNFUP.C, S@DYNCFY.C, S@DYNLST.C, S@DYNFRE.C

Procedure Name: dynBuildTxtConfigRec, dynVerifyAndBuildRec, dynCmpConfigEntries, dynFreeStr

---

EZZ6299I Incorrect number of arguments for *(pwtokey | pwchange)*

Explanation: The pwtokey or pwchange function was invoked with too few or too many arguments

System Action: pwtokey or pwchange function ends.

User or Operator Response: Invoke the command with the correct number of arguments. The **-?** option can be used to display the usage statement.

System Programmer Response: None.

Source Data Set: EZASNPWT.C

Procedure Name: parse_options

EZZ6300I *(pwtokey | pwchange)* **value for option** *option* **is not valid.**

Explanation: The pwtokey or pwchange function was invoked with a value that was not valid for the specified option or the value was omitted.

System Action: pwtokey or pwchange function ends.

User or Operator Response: Invoke command correctly. The **-?** option can be used to display the usage statement.

System Programmer Response: None.

Source Data Set: EZASNPWT.C

Procedure Name: parse_options

EZZ6301I Passwords for *(pwtokey | pwchange)* **must be at least** *number* **characters long.**

Explanation: The pwtokey or pwchange function was invoked with a password that was too short.

System Action: pwtokey or pwchange function ends.

User or Operator Response: Invoke command correctly. The **-?** option can be used to display the usage statement.

System Programmer Response: None.

Source Data Set: EZASNPWT.C, S@PWCHNG.C

Procedure Name: parse_options

EZZ6302I *(EngineID | Host name or IP address)* **for** *(pwtokey | pwchange)* **is not valid**

Explanation: The pwtokey or pwchange function was invoked with an agent identification that was not valid.

System Action: pwtokey or pwchange function ends.

User or Operator Response: Invoke command correctly. The **-?** option can be used to display the usage statement.

System Programmer Response: None.

Source Data Set: EZASNPWT.C, S@PWCHNG.C

Procedure Name: parse_options

EZZ6303I pwchange usage statement

Explanation: The pwchange usage statement is displayed when pwchange is invoked with the **-?** option.

System Action: pwchange ends after displaying the usage statement.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: S@PWCHNG.C

Procedure Name: do_usage

EZZ6304I SNMP agent: Input in *function* is not valid.

Explanation: The indicated SNMP agent function was invoked with input that is not valid.

System Action: If the error occurs during SNMP agent initialization, the SNMP agent will terminate. If the error occurs while updates are being written to the SNMPD.CONF file, the SNMP agent will terminate. If the error occurs while the SNMP agent is processing a SET request, the SNMP agent fails the SET request.

User or Operator Response: Collect SNMP agent traces. If the error occurs at SNMP agent initialization, collect traces by restarting the SNMP agent with the **-d** parameter set to 255. If the error occurs as a result of an SNMP SET command, activate traces by using the MVS MODIFY command to set the trace level to 255. Provide the traces and the SNMP agent configuration file, SNMPD.CONF, to the system programmer.

System Programmer Response: Contact the IBM Software Support Center with the traces and the SNMPD.CONF file.

Source Data Set: S@DYNCFY.C, S@DYNLST.C

Procedure Name: dynUpdateConfig, dynAddToConfigList, dynDeleteFromConfigList, dynModifyConfigList, dynCheckString

---

EZZ6305I SNMP agent: String contains non-printable US ASCII characters

Explanation: The SNMP agent dynamic configuration function received an entry that contained non-US ASCII UTF8 characters. Non-US ASCII UTF8 characters are not supported in the SNMP agent's SNMPD.CONF configuration file.

System Action: If the error occurs during SNMP agent initialization, the SNMP agent will terminate. If the error occurs while the SNMP agent is processing a SET request, the SNMP agent fails the SET request.

User or Operator Response: None.

System Programmer Response: Repeat the configuration step using only US ASCII characters.

Source Data Set: S@DYNCFY.C

Procedure Name: dynCheckString

EZZ6306I SNMP agent: Cannot find entry to delete from the dynamic configuration list.

Explanation: The SNMP agent was invoked to remove a configuration entry from the list representing the SNMPD.CONF entries, but the entry to be deleted was not found.

System Action: The SNMP agent fails the SET request.

User or Operator Response: Collect SNMP agent traces. Activate traces by using the MVS MODIFY command to set the trace level to 255. Repeat the failing SNMP SET command. Provide the traces and the SNMP agent configuration file, SNMPD.CONF, to the system programmer.

System Programmer Response: Contact the IBM Software Support Center with the traces and the SNMPD.CONF file.

Source Data Set: S@DYNLST.C

Procedure Name: dynDeleteFromConfigList

EZZ6307I SNMP agent: Error reopening *filename* file for writing.

Explanation: The SNMP agent dynamic configuration processing attempted to reopen the SNMPD.CONF configuration file to write updates made by SET requests to the SNMPD.CONF file. However, an error was encountered while opening the file.

System Action: The SNMP agent terminates.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why the file could not be opened, correct the problem and restart the SNMP agent.

Source Data Set: S@DYNFUP.C

Procedure Name: dynUpdateConfigFile

EZZ6308I SNMP agent: Default *entry type* **entry definition not redefined.**

Explanation: The SNMP agent was attempting to set up the default configuration but found that a conflicting entry of the specified type was already defined. The preexisting entry definition is used instead of the default definition.

System Action: The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: Verify that the SNMP agent is configured as desired. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about the default SNMP agent configuration.

Source Data Set: EZASNAC3.C

Procedure Name: configure_v3_default

EZZ6309I Passwords for *(pwtokey | pwchange)* **must be no more than** *number* **characters long.**

Explanation: The pwtokey or pwchange function was invoked with a password that was too long.

System Action: pwtokey or pwchange function ends.

User or Operator Response: Invoke command correctly. The **-?** option can be used to display the usage statement.

System Programmer Response: None.

Source Data Set: EZASNPWT.C, S@PWCHNG.C

Procedure Name: parse_options

EZZ6310I *(pwtokey | pwchange)* **option** *option* **is not valid.**

Explanation: The pwtokey or pwchange function was invoked with an option that was not valid.

System Action: pwtokey or pwchange function ends.

User or Operator Response: Invoke command correctly. The **-?** option can be used to display the usage statement.

System Programmer Response: None.

Source Data Set: EZASNPWT.C, S@PWCHNG.C

Procedure Name: parse_options

EZZ6311I SNMP agent terminating, cannot update configuration file *filename***.**

Explanation: The SNMP agent was attempting to update the configuration file after a dynamic configuration change, but the file update failed. Possible causes may include insufficient authority to write the file or insufficient space

System Action: The agent ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: See the syslog daemon trace output for more information about why the file update failed. Correct the problem and restart the SNMP agent.

Source Data Set: EZASNAA3.C

Procedure Name: main

EZZ6312I SNMP agent: Attempt to update configuration file failed. Agent continues.

Explanation: The SNMP agent was attempting to update the configuration file with dynamic configuration changes, but the file did not exist.

System Action: The agent continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: The agent was initialized with only community based security. Dynamic configuration is supported only when the SNMPD.CONF file exists. Any dynamic configuration changes attempted will be lost when the SNMP agent is restarted. If the ability to make dynamic configuration changes is desired, configure the SNMP agent using SNMPV3 security. Refer to the *z/OS Communications Server: IP Configuration Guide* for information about SNMP agent configuration.

Source Data Set: S_DYNFUP.C

Procedure Name: main

EZZ6313I *interval value* **is an incorrect interval value, using** *default interval value*

Explanation: When the agent was started, a -i argument specified a value that was not a valid interval. The value must be between 0 and 10.

System Action: The agent terminates.

User or Operator Response: The user should restart the agent with a valid interval value. For more information, see the *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZASNAC3.C

Procedure Name: snmp_config_parse_options

EZZ6314I *(pwchange | pwtokey)* **is a UNIX System Services command and must be issued from an OS/390 shell**

Explanation: The pwchange / pwtokey command was not issued from an OS/390 shell.

System Action: The pwchange / pwtokey command terminates.

User or Operator Response: Reissue the command from an OS/390 shell. For information about the pwchange / pwtokey command, see the *z/OS Communications Server: IP User's Guide and Commands* and *z/OS Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EZASNPWT.C, S@PWCHNG.C

Procedure Name: main

EZZ6315I SNMP agent: line number *number* **contains a keyword value that is incorrect:** *keyword*

Explanation: The indicated keyword value in the SNMP agent configuration file is incorrect. For example, if UTF8 characters are used, the number of characters following the > sign might be incorrect.

System Action: The entry is ignored. The SNMP agent continues initialization.

User or Operator Response: None.

System Programmer Response: Correct the entry in the configuration file and restart the SNMP agent. Refer to the *z/OS Communications Server: IP Configuration Reference* for configuring the SNMP agent.

Source Data Set: EZASNASO.C

Procedure Name: xlateUTF8String

EZZ6316I Refused new DPI subagent connection due to authorization failure - SAF RC is *safrc* security RC is *return_code* security reason code is *reason_code*

Explanation: The SNMP agent refused the connection of a new subagent because of a security product authorization failure. If the security product profile EZA.SNMPAGENT.sysname.tcpprocname was defined for this SNMP agent, then the agent will refuse connections from a subagent for one of the following reasons:

- The subagent is not associated with the same TCP/IP stack as the agent. This includes subagents associated with other TCP/IP stacks on the same MVS image or sysplex, or remote subagents.
- The subagent is associated with the same TCP/IP stack as the agent but the agent is unable to obtain the security product user ID associated with the subagent.
- The subagent is associated with the same TCP/IP stack as the agent but the subagent's security product user ID is not authorized to the agent's security product profile.

safrc is the hexadecimal SAF return code.

return_code is the hexadecimal return code from the security product.

reason_code is the hexadecimal reason code from the security product.

System Action: The SNMP agent continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Locate the specified *return_code* and *reason_code* in your installed host security product documentation to determine the cause of the error. If you are using the z/OS Security Server (RACF) as your security product, the *return_code* and *reason_code* are documented in the *z/OS Security Server RACROUTE Macro Reference* in the section about return and reason codes for the RACROUTE REQUEST=AUTH function. Your installed host security product might have issued messages regarding the failure.

For additional diagnostic information, re-create the problem with the SNMP agent traces at level **-d 128**. Earlier agent trace messages in the syslog daemon trace output might indicate the cause of the authorization failure. See the *z/OS Communications Server: IP Configuration Guide* for more information about the agent security product profile.

Source Data Set: EZASNAA3

Procedure Name: main

EZZ6317I Configuration of MIB object *MIB_variable_name* accepted but will not be allowed in future releases

Explanation: A statement configuring MIB object *MIB_variable_name* was found in the OSNMPD.DATA file. Configuration of the object is currently accepted but will not be allowed in future releases. The default value is recommended.

MIB_variable_name is the name of the MIB variable.

System Action: The SNMP agent accepts the configuration of the MIB object.

User or Operator Response: None.

System Programmer Response: To prepare for future releases, update the OSNMPD.DATA file to remove the setting for this MIB object.

Source Data Set: set_mib_install_defaults

Procedure Name: EZASNAA3.C

EZZ6351I Abend detected.

Explanation: This message indicates that an internal programming error caused the command to abnormally terminate.

System Action: The command ends.

User or Operator Response: Collect debug information using the DEBUG/-d option and forward the results to your system programmer or administrator for resolution. If you invoked the otracert/traceroute command, you can save

debug information by redirecting it to a file using the '>' operator.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: abndhand

EZZ6352I IPv4-mapped IPv6 address not supported

Explanation: You specified an IPv4-mapped IPv6 address that is not supported by the command.

System Action: The command ends.

User or Operator Response: Correct the IP address specification and reissue the command. For information about the command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: getHost, getSrcip, getIntf

EZZ6353I Option *option* **ignored, setsockopt() failed:** *description* **(***return_code***/***reason_code***)**

Explanation: The command tried to do a setsockopt() for option *option* but the setsockopt() failed.

option is the option for which the setsockopt() was attempted.

description describes the Return Code.

return_code is the decimal UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

reason_code is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The command continues but the option *option* is ignored.

User or Operator Response: Contact the system programmer.

System Programmer Response: If the reason code was set by the TCP/IP stack, re-create the problem with the DEBUG/-d option specified to obtain a trace. Contact the IBM Support Center with the trace output. If the reason code was not set by the TCP/IP stack, contact the product that set the reason code for assistance.

Source Data Set: EZACDTRT

Procedure Name: openSock

EZZ6354I *socket_operation* **error detected** *description* **(***return_code***/***reason_code***)**

Explanation: The command was unable to perform the indicated socket operation.

socket_operation is the socket function that failed.

description describes the Return Code.

return_code is the UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

return_code is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The command ends.

User or Operator Response: Collect debug information using the DEBUG/-d option and forward the results to your system programmer or administrator for resolution. If you invoked the otracert/traceroute command, you can save debug information by redirecting it to a file using the '>' operator.

System Programmer Response: None.

Source Data Set: EZACDTRT, EZACDTR6

Procedure Name: openSock, processTracerte, recvPacket, processTracerte6

EZZ6355I *option* **option ignored for IPv6 destination**

Explanation: The indicated option is not supported for IPv6 destinations.

option is the command option that is not supported.

System Action: The option is ignored and processing continues.

User or Operator Response: To avoid receiving this message when using IPv6 destinations, specify only those command options that are supported for IPv6. For more information about the command options, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: trtMain

EZZ6356I **Unable to retrieve HOME list for** *tcpname*- *description* **(***return_code***/***reason_code***)**

Explanation: The command was unable to obtain the HOME list from TCP/IP instance *tcpname*. If the TCP/IP instance name is TCPIP this might mean that the command is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used. The command tries to obtain the HOME list when the INTF/-i option is specified.

tcpname is the name of the TCP/IP stack from which Traceroute tried to obtain the HOME list.

description describes the meaning of the Return Code.

return_code is the decimal UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

reason_code is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The command ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: If the reason code was set by the TCP/IP stack, re-create the problem with CTRACE options UDP, PFS, and IOCTL active. Contact the IBM Support Center with the CTRACE. If the reason code was not set by the TCP/IP stack, contact the product that set the reason code for assistance.

Source Data Set: EZACDTRT

Procedure Name: getIfIndex

EZZ6357I **Value** *value* **for option** *option* **is not an integer**

Explanation: You specified a non-numeric value when a numeric value was expected.

System Action: The command will use the default value for the option and continue processing.

User or Operator Response: Cancel execution and correct option if default is not acceptable. For more information see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDUTR, EZACDTTR

Procedure Name: optArgCh, tokValChk, parsUTR, parsTTR

EZZ6358I The value of *option* must be between *minvalue* and *maxvalue*.

Explanation: You specified an incorrect *option* value.

System Action: The command will use the default value for the option and continue processing.

User or Operator Response: Cancel execution if accepting default is not desired.

System Programmer Response: None.

Source Data Set: EZACDTRT, EZACDTTR, EZACDUTR

Procedure Name: optArgCh, tokValChk, trtMain

EZZ6359I Incorrect *option* option value specified

Explanation: You specified an incorrect option value.

option is the command option for which the incorrect value was specified.

System Action: The command ends.

User or Operator Response: Specify the command again with a correct value for the indicated option. For the INTF/-i or SRCIP/-s options, the IP address or interface must be of the same address family as the destination.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: getSrcip, getIfIndex

EZZ6361I Extraneous option '*option*'.

Explanation: You specified an extraneous option.

System Action: The command ends.

User or Operator Response: Correct the syntax of the incorrect option and reissue the command. For information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTTR, EZACDUTR

Procedure Name: parsTTR, parsUTR

EZZ6362I Unknown option '*option*'.

Explanation: You specified an unknown option.

System Action: The command ends.

User or Operator Response: Check the unknown option for misspellings or other problems. Correct the option and reissue the command. For information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTTR, EZACDUTR

Procedure Name: parsTTR, parsUTR

EZZ6363I Host name too long '*name*'.

Explanation: The field interpreted as the host name is too long. Host name must be less than or equal to 255 characters in length.

System Action: The command ends.

User or Operator Response: Correct the host name and reissue the otracert command. For information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTTR, EZACDUTR

Procedure Name: parsTTR, parsUTR

EZZ6364I Allocation of probe failed - *packetSize***.**

Explanation: This error should not occur and indicates that the command was unable to allocate the probe packet buffer.

System Action: The command ends.

User or Operator Response: Retry the command with a smaller packetSize selection. For information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: trtMain

EZZ6365I Host name or address not entered

Explanation: A requested host name or IP address was missing.

System Action: The command ends.

User or Operator Response: Reissue the command with the host identification included.

System Programmer Response: None.

Source Data Set: EZACDTTR, EZACDUTR

Procedure Name: parsTTR, parsUTR

EZZ6367I Mismatched IP address type values

Explanation: The command found a mismatch in IP address type (that is, IPv4 or IPv6) between some of the IP address values specified. IP address type can be specified with the ADDRTYPE/-A option; or by providing an IP address as the destination host, SRCIP/-s option value, or INTF/-i option value. If a LINK or INTERFACE name was specified for the INTF/-i option value, the command will verify that the LINK or INTERFACE is of the same IP address type as the host or other IP address options specified.

System Action: The command ends.

User or Operator Response: Verify that the values specified for destination host, and the ADDRTYPE/-A, SRCIP/-s, or INTF/-i options, are all of the same IP address type, IPv4 or IPv6.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: getSrcip, getIntf, getIfIndex, getHost

EZZ6370I Failure detected in closing socket *socket* **(***return_code***/***reason_code***).**

Explanation: close() error

socket is the socket descriptor of the socket for which close() failed.

return_code is the UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

reason_code is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The command ends.

User or Operator Response: Collect debug information using the DEBUG/-d option and forward the results to your

system programmer or administrator for resolution. If you invoked the otracert/traceroute command, debug information can be saved by redirecting it to a file using the '>' operator.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: clsSock, getIfIndex

EZZ6371I Unable to open message catalog 'trtemsg.cat' - *description* **(***return_code***/***reason_code***)**

Explanation: The command was unable to open the message catalog "trtemsg.cat" in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be "NLSPATH=/usr/lib/nls/msg/%L/%N".

description describes the return code.

return_code/*reason_code* is the UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

reason_code is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The command will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *z/OS C/C++ Run-Time Library Reference* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in *z/OS UNIX System Services Programming Tools*. If the default messages are acceptable, no action is necessary.

Source Data Set: EZACDTTR, EZACDUTR

Procedure Name: main

EZZ6372I sigaction() failed for *signal* **:** *reason*

Explanation: The command encountered an error attempting to set up the signal handler for the signal specified by *signal*.

reason is the error returned by the C run-time library for the failing sigaction() call. If the signal handler is not correctly enabled, the command 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 may not be properly cleaned up.

SIGTERM
handler controls cleanup of resources during termination.

SIGPIPE
handler allows the command to detect when the connection to the command was terminated.

SIGINT handler controls cleanup of resources when the user requested that the command stop processing and end.

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: EZACDTTR, EZACDUTR

Procedure Name: main

EZZ6373I Unknown host *name*

Explanation: The host name specified could not be resolved to an IP address. If the Resolver could not resolve the host name to an IP address, message EZZ6366I should have been issued prior to this message. See message EZZ6366I for more information.

System Action: The command ends. Verify that host name resolution was configured correctly.

User or Operator Response:

- Correct the syntax of the host name and reissue the command. For information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.
- Check that the specified host name is valid. If the host name looks correct, check with your system programmer to verify the host address.
- Use the IP address, if it is known.

System Programmer Response: If message EZZ6366I was issued, follow the recommended actions for that message. Otherwise, re-create the problem with the DEBUG/-d option to obtain a trace. Contact the IBM Support Center with the trace output.

Source Data Set: EZACDTRT

Procedure Name: getHost

EZZ6374I Execution was interrupted

Explanation: You requested that the command stop processing and end. The TSO TRACERTE command can be interrupted by using the PA1 or ATTN key. The UNIX otracert/traceroute command can be interrupted by entering the ESC character plus C or c.

System Action: The command execution is interrupted. The command ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: inthand

EZZ6375I Found parenthesis but no options

Explanation: You specified a parenthesis without specifying any options.

System Action: The command ends.

User or Operator Response: Specify options or remove the parenthesis and reissue the TRACERTE command. For information about the TRACERTE command, see the *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDTTR

Procedure Name: parsTTR

EZZ6376I Execution was canceled

Explanation: None.

System Action: The command was stopped by either user or operator action.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: termhand

EZZ6377I Could not establish affinity with *tcpipname* **(***return_code***/***reason_code***)**

Explanation: The command attempted to use the setibmopt() socket call to associate itself with the TCP/IP instance *tcpipname*. The *tcpipname* in this message will be one of the following:

- The name specified on the TCP/-a option.
- The TCPIPjobname value retrieved from the resolver configuration data set because the INTF/-i option was specified.
- 'INET' if the INTF/-i option was specified without the TCP/-a option and the __iptcpn function was unable to find a default TCP/IP name.

This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifer' format of the MVS Start command was used) of the TCP/IP instance to which the user wishes to direct the otracert command. The setibmopt call failed with the displayed *return_code* and *reason_code*.

System Action: The command ends.

User or Operator Response: The most probable cause of the error is that the TCP/IP instance name was not defined correctly to UNIX System Services. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure UNIX System Services. Ensure that the TCP/IP started procedure name (or identifier if the 'S member.identifier' format of the MVS Start command was used) matches the SUBFILESYSTYPE NAME. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their names. If none of the above error conditions exist contact your system programmer.

System Programmer Response: Correct the error indicated by *return_code* and *reason_code*.

Source Data Set: EZACDTRT

Procedure Name: trtMain

EZZ6378I Executing under INET configuration. The *option* **option is ignored.**

Explanation: The command was issued in a UNIX System Services environment configured for INET. In an INET configuration, there can only be one TCP/IP (for example, AF_INET type) stack connected to UNIX System Services. In this case, the command continues but ignores the TCP/-a option.

System Action: The command continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: trtMain

EZZ6379I *'value'* **specified for** *option* **must be** *length* **characters or less**

Explanation: You specified an incorrect *option* value.

System Action: The command ends.

User or Operator Response: Specify a *value* less than or equal to requested characters and reissue the otracert command.

System Programmer Response: None.

Source Data Set: EZACDTRT, EZACDTTR, EZACDUTR

Procedure Name: getIntf, parsTTR, parsUTR

EZZ6380I Unable to open *socket_operation* **socket,** *tcpipname* **is not active**

Explanation: The command attempted to open a socket for *socket_operation* processing. The open of the socket failed because the TCP/IP stack *tcpipname* was not active.

System Action: The command ends.

User or Operator Response: Determine why the TCP/IP stack is not active. The *tcpipname* in this message will be one of the following:

- The name specified on the TCP/-a option.
- The TCPIPjobname value retrieved from the resolver configuration data set because the INTF/-i option was specified.
- TCPIP if the command is executing in an INET environment.

When neither the TCP/-a option nor the INTF/-i option is specified and UNIX System Services is configured for CINET, the CINET Prerouter selects the TCP/IP stack to which to route the request.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: openSock

EZZ6381I Missing value after *option* option.

Explanation: You specified *option* without a value.

System Action: The command ends.

User or Operator Response: Specify a value for *option* that is between the accepted minimum and maximum values and reissue the command.

System Programmer Response: None.

Source Data Set: EZACDTTR, EZACDUTR

Procedure Name: parsTTR, tokValChk, parsUTR

EZZ6382I otracert is a UNIX Shell command and must be issued from a UNIX shell.

Explanation: The otracert command was not issued from a UNIX shell.

System Action: The command ends.

User or Operator Response: Reissue the command from a UNIX shell. For information about the command, see *z/OS Communications Server: IP System Administrator's Commands*.

System Programmer Response: None.

Source Data Set: EZACDUTR

Procedure Name: main

EZZ6384I Could not determine the TCPIPjobname, using default of 'INET'

Explanation: The command invoked the __iptcpn() function to retrieve the resolver supplied TCPIPjobname but no name was returned. A default value of 'INET' will be used for TCPIPjobname.

System Action: The command continues.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for the command to communicate with a particular stack, the TCPIPjobname should be set in the appropriate resolver configuration file or data set. Make sure that the TCPIPjobname statement in the appropriate resolver configuration file or data set is correct and resubmit the command. For more information about the search order for locating the resolver configuration file or data set, see the *z/OS Communications Server: IP Configuration Guide*. The DISPLAY TCPIP operator command can be used to display all started TCP/IP instances and their jobnames. The TCP/-a command options can be used to explicitly select a TCP/IP instance by specifying its jobname.

Source Data Set: EZACDTRT

Procedure Name: trtMain

EZZ6385I The interface *interface* was not found in the HOME list for *tcpname*

Explanation: The interface *interface* was specified on the INTF/-i option but was not defined to the TCP/IP instance named *tcpname*. If the TCP/IP instance name is TCPIP this could mean that the command is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used.

System Action: The command ends.

User or Operator Response: Verify that the value specified for the INTF/-i option is defined to the TCP/IP instance named *tcpname*. You can use the Netstat HOME/-h command to verify the interfaces defined to a TCP/IP instance. Reissue the command with a valid value for the INTF/-i option.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: getIfIndex

EZZ6386I Interface *interface* for *tcpname* can not be specified for the *option* option.

Explanation: The command was invoked with the INTF/-i option to specify the interface name or IP address on which to send out the probe packets. The interface *interface* was found in the HOME list of the TCP/IP instance name *tcpname*, but the interface type cannot be specified for the INTF/-i option.

If the TCP/IP instance name is TCPIP this could mean that the command is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used.

System Action: The command ends.

User or Operator Response: Ensure that the interface specified for the INTF/-i option is not a VIPA interface or a loopback interface and reissue the command.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: getIfIndex

EZZ6387I More than one IP address *ipaddr* found for *tcpname*.

Explanation: The command was invoked with an IP address specified for the INTF/-i option but there is more than one interface defined to the TCP/IP instance *tcpname* with IP address *ipaddr*. If the TCP/IP instance name is TCPIP this could mean that the command is executing in an INET environment and is not aware of the actual name of the TCP/IP instance being used.

System Action: The command ends.

User or Operator Response: If you have more than one interface with the same IP address, Traceroute cannot determine on which interface you want the outbound packets sent. Reissue the command specifying the interface name on the INTF/-i option to identify the interface to Traceroute.

System Programmer Response: None.

Source Data Set: EZACDTRT

Procedure Name: getIfIndex

EZZ6388I *function* failed due to error *h_errno*

Explanation: The function *function* issued by the command failed with error *h_errno*. Possible causes of the problem are an error in specifying the configuration file for the Resolver, or that the Resolver address space was not started.

function is the C/C++ Run-Time function that failed.

h_errno is one of the following errors:

1 HOST_NOT_FOUND

2 TRY_AGAIN

3 NO_RECOVERY

4 NO_DATA

System Action: For some function failures, the command ends. For other function failures, the command continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the Resolver address space is started. See the *z/OS Communications Server: IP Configuration Guide* for information about starting the Resolver address space. If this does not resolve the problem, correct the error indicated by the h_errno value. See the *z/OS C/C++ Run-Time Library Reference* for a description of the *h_errno* for the function that failed.

Source Data Set: EZACDTRT

Procedure Name: trtMain

EZZ6452I named starting. *version information*

Explanation: The domain name server is starting.

System Action: The domain name server begins initializing, reading its configuration information and allocating any required system resources.

User or Operator Response: Refer to *z/OS Communications Server: IP Configuration Reference* for information about the domain name server.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6453E named: socket(SOCK_STREAM) failed: *error information*

Explanation: The domain name server could not create its TCP socket.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the socket() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6455E setsockopt(vs, reuseaddr) failed: *error information*

Explanation: The domain name server could not set its TCP socket as reusable.

System Action: The domain name server tries to re-create and reinitialize its TCP socket. If the socket cannot be created, the name server ends.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the setsockopt() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6457E named: there may be a name server already running

Explanation: The domain name server could not bind to its TCP socket. The bind() function failed with an indication the specified local address and port is already in use. There is probably already another domain name server active. The name server uses port 53, unless overridden when started, which is reserved for DNS usage.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Determine if another name server is running, and take appropriate action. If another application is using port 53 (or the port you wish to use for the domain name server), stop the application and restart the name server.

See *IBM Open Class Library Reference, Vol. 1* for more information about the bind() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6458I named ending

Explanation: The domain name server encountered an unrecoverable problem. Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6459E named: bind(vs, [*name*].*port*) failed: *error information*

Explanation: The domain name server could not bind to its TCP socket and port.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the bind() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6461E named: listen failed: *error information*

Explanation: The domain name server could not prepare to receive incoming TCP connections. The listen() function failed for its TCP socket.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the listen() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6464I Usage: named [-d #] [-q] [-r] [-p port[/localport]] [[-b] bootfile][-t seconds]

Explanation: A short description of the command line parameters used to start the domain name server.

System Action: The domain name server displays the information and ends.

User or Operator Response: Use this information to start the domain name server correctly. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information.

System Programmer Response: Assist the user if necessary.

Source Data Set: NS_MAIN

Procedure Name: usage

EZZ6467E named-xfer: name server zone transfer usage error: *type of error*

Explanation: The domain name server encountered an internal software error while processing a zone transfer. A should-not-occur condition was detected. The name server is a secondary or caching name server for the requested zone.

The types of errors that may have been encountered are:

- unrecognized argument given when named-xfer was loaded,
- no domain specified on zone transfer,
- no file specified to hold zone transfer information,
- file type incorrect, or
- not enough arguments passed to named-xfer.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. If possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Note: Some domain name system publications describe how to use the named-xfer program directly (mostly when debugging zone transfer problems). If this error occurred while manually starting the named-xfer program, you probably did not specify the correct command line options. Check how to invoke the named-xfer program and issue the command correctly. Never explicitly start the named-xfer program in a production environment.

Source Data Set: NAMED@XF

Procedure Name: usage

EZZ6468E named-xfer: name server zone transfer failed, named-xfer ending

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

If you cannot determine the cause of the problem or the problem persists, contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. In addition include copies of the zone files for the name servers involved in the problem. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6469I *statistics information*

Explanation: The domain name server is displaying a subset of its statistics.

System Action: The domain name server continues normal processing.

User or Operator Response: None.

System Programmer Response: None. See *z/OS Communications Server: IP Diagnosis* for information about how to read and interpret the statistics.

Source Data Set: NS_STATS

Procedure Name: ns_logstats

EZZ6470I *sysloginfo*: **query(***query name***)** *complaint* **(***name***:***address***)**

Explanation: The domain name server could not forward a query request for the indicated name to other name servers for resolution. The name server is trying to forward the request to other name servers because it did not have the information needed to answer the query. It did not forward the request because it detected a problem in the request that may have caused network problems.

System Action: The domain name server sends back a response indicating an error occurred while processing the query. The name server continues normal processing.

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

System Programmer Response: No immediate action is required unless the error persists. The *complaint* information in this message identifies the error condition. Some common complaints are:

contains our address
: The query contains invalid forwarding information. The forwarding information includes the local name server.

forwarding loop
: The domain name servers in the network are incorrectly defined or are behaving incorrectly. The local name server determined that it should forward the request to the TCP/IP host that sent the request. This should not occur, since it can cause a forwarding loop for the query between two name servers.

Reconfigure the name servers in your network if you suspect a configuration problem.

If you cannot determine the cause of the problem, contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. In addition include copies of the zone files for the name servers involved in the problem. If possible, dump the domain name server's resource data and statistics and provide that information as well. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP User's Guide and Commands* for information about a dump of the name server's resource data and to use its debugging options.

Source Data Set: NS_FORW

Procedure Name: nslookupComplain

EZZ6471I name server zone transfer failed, named-xfer ending received inverse query from *hostname* **for** *address* **(type=***type***)**

Explanation: The domain name server received a query request for the indicated information. The query is for the host name associated with a given IP address. This message is only displayed when query logging is enabled.

System Action: The name server processes the query request.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_REQ

Procedure Name: req_iquery

EZZ6472I received query from *hostname* **for** *domainname* **(type=***type***)**

Explanation: The domain name server received a query request for the indicated information. This message is only displayed when query logging is enabled.

System Action: The name server processes the query request.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6473E named-xfer: *file name***:** *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not store the received information in the temporary zone file.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the domain name server has write access to the directory and files and that there is enough file space available to hold the received resource information. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the file write functions (for example, putc() and fprintf()) and their possible error values.

Source Data Set: NAMED@XF

Procedure Name: print_output

EZZ6474E named: fork failed: *error information*

Explanation: The domain name server could not make itself a daemon process. The name server, after initializing, runs in the background. The fork() function failed for the name server process.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the fork() function and its possible error values.

Source Data Set: PLATFORM

Procedure Name: fork_it

EZZ6475I named: ready to answer queries.

Explanation: The domain name server finished loading its resource information and is now available to respond to query requests.

System Action: The domain name server continues normal processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN, DB_RELOA

Procedure Name: main, db_reload

EZZ6476I Return from getdtablesize() > FD_SETSIZE

Explanation: The number of possible sockets (file descriptors) supported by the system (as defined by MAXFILEPROC in BPXPRMxx) exceeds the maximum allowed by the name server.

System Action: The domain name server continues normal processing. The maximum number of sockets supported by the name server is adjusted to the hard coded maximum of 2048.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6477E select failed: *error information*

Explanation: The domain name server encountered an error while waiting for a request. The select() function failed for an unexpected reason.

System Action: The domain name server tries to recover and continue normal processing. It will pause for 60 seconds and then continue. During that 60 seconds, the name server will not be answering query requests.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the select() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6479I accept failed: *error information*

Explanation: The domain name server encountered an error while trying to accept an incoming TCP connection. The accept() function failed.

System Action: The domain name server tries to recover and continue normal processing. The TCP connection is not accepted, but the remote host should recover. DNS resolvers will time out their TCP connections and retry the connection or try and connect to a different name server.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the accept() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6480I ioctl failed(*socket*, FIONBIO): *error information*

Explanation: The domain name server encountered an error while trying to accept an incoming TCP connection. The error occurred trying to change the mode of the incoming connection to non-blocking. The ioctl() function failed on the socket descriptor returned from the accept().

System Action: The domain name server tries to recover and continue normal processing. The TCP connection is closed, but the remote host should recover. DNS resolvers will detect the connection failure and retry the connection or try and connect to a different name server.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the accept() and ioctl() functions and their possible error values.

Source Data Set: NS_MAIN

Procedure Name: main, opensocket

EZZ6481E Zone '*zone name*' declared more than once

Explanation: The domain name server detected a syntax error in the boot file. The boot file contains multiple definitions for the indicated zone.

System Action: The domain name server ignores the extraneous zone definition and continues normal processing. The resources defined in the zone file associated with the duplicate definition are not loaded. The name server will not respond to queries for those resources.

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

System Programmer Response: Remove the duplicate zone definitions from the boot file and restart or reload the domain name server. See the *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6482I getsockopt(rfd, IP_OPTIONS) failed: *error information*

Explanation: The domain name server encountered an error while trying to accept an incoming TCP connection. The error occurred trying to obtain some of the IP options associated with the incoming connection. The getsockopt() function failed for the socket descriptor returned from the accept().

System Action: The domain name server tries to recover and continue normal processing. The TCP connection is closed, but the remote host should recover. DNS resolvers will detect the connection failure and retry the connection or try and connect to a different name server.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the accept() and getsockopt() functions and their possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6483W received IP_OPTIONS from [*address*].*port* (ignored)

Explanation: The domain name server was able to accept an incoming TCP connection, but there already existed some IP options set for the newly created socket.

System Action: The domain name server ignores the existing IP options, clearing any existing options before continuing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6484I setsockopt(!IP_OPTIONS) failed: *error information*

Explanation: The domain name server encountered an error while trying to accept an incoming TCP connection. The error occurred trying to reset the existing IP options for the incoming connection. The setsockopt() function failed for the socket descriptor returned from the accept().

System Action: The domain name server tries to recover and continue normal processing. The TCP connection is closed, but the remote host should recover. DNS resolvers will detect the connection failure and retry the connection or try and connect to a different name server.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the accept() and setsockopt() functions and their possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6486E setsockopt(rfd, KEEPALIVE) failed: *error information*

Explanation: The domain name server encountered an error while trying to accept an incoming TCP connection. The error occurred trying to specify that the connection should remain active during periods of inactivity. The setsockopt() function failed for the socket descriptor returned from the accept().

System Action: The domain name server tries to recover and continue normal processing. The TCP connection is closed, but the remote host should recover. DNS resolvers will detect the connection failure and retry the connection or try and connect to a different name server.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the accept() and setsockopt() functions and their possible error values.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6487E named: get interface configuration failed: *error information*

Explanation: The domain name server could not prepare to receive incoming datagrams (query requests). The name server encountered an error obtaining the hardware interface information from the system. The ioctl(type=SIOCGIFCONF) function failed for the TCP socket.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *z/OS C/C++ Run-Time Library Reference* for more information about the ioctl() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: getnetconf

EZZ6488I get interface addr: *error information*

Explanation: An error occurred while getting an ioctl command. The name server may not listen on all addresses defined.

System Action: The name server continues.

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

System Programmer Response: Use the error information in this message to determine the cause of this error and respond as indicated.

See *IBM C/C++ for MVS/ESA Library Reference* for more information about the ioctl() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: getnetconf

EZZ6489C named: getnetconf: storage allocation failed: *error information*

Explanation: The domain name server could not expand its internal cache. It did not have the storage needed to hold all the resource definitions and information known by the name server.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (calloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the calloc() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: getnetconf

EZZ6490I get netmask failed: *error information*

Explanation: The domain name server encountered a problem while preparing to receive incoming datagrams (query requests). The name server opens and configures a datagram socket for every local IP address. The domain name server was unable to query the network mask associated with a local IP address. The ioctl(type=SIOCGIFNETMASK) function failed for the UDP socket.

System Action: The domain name server continues processing, associating a generic class A, class B, or class C

network mask with the local IP address. The type of mask used is the same as the class of the local IP address.

User or Operator Response: None.

System Programmer Response: No action is required if the default network mask is acceptable.

Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the ioctl() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: getnetconf

EZZ6491I get interface flags failed: *error information*

Explanation: The domain name server encountered a problem while preparing to receive incoming datagrams (query requests). The name server opens and configures a datagram socket for every local IP address. The domain name server was unable to query interface information associated with one of the local IP addresses. The ioctl(type=SIOCGIFFLAGS) function failed for the UDP socket.

System Action: The domain name server continues processing, using default actions to recover from the inability to determine certain interface information:

- the name server does not check to see if the local IP address is a loopback address,
- the name server does not set the configured point-to-point IP address if the local interface is point-to-point (the local IP address is used)

User or Operator Response: None.

System Programmer Response: No action is required if the default actions described above are acceptable.

Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the ioctl() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: getnetconf

EZZ6492I get point-to-point local IP addr failed: *error information*

Explanation: The domain name server encountered a problem while preparing to receive incoming datagrams (query requests). The name server opens and configures a datagram socket for every local IP address. The domain name server was unable to retrieve the point-to-point interface information associated with a point-to-point local IP addresses. The ioctl(type=SIOCGIFDSTADDR) function failed for the UDP socket.

System Action: The domain name server continues processing, using default actions to recover from the inability to obtain the information. The name server uses the local IP address instead of the configured point-to-point IP address.

User or Operator Response: None.

System Programmer Response: No action is required if the default action described above is acceptable.

Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the ioctl() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: getnetconf

EZZ6493E named: socket(SOCK_DGRAM) failed: *error information*

Explanation: The domain name server encountered an error while preparing to receive incoming datagrams (query requests). The name server opens and configures a datagram socket for every local IP address. The socket() function failed, preventing the creation of one of the UDP sockets.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the socket() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: opensocket

EZZ6494W setsockopt(reuseaddr) failed: *error information*

Explanation: The domain name server encountered an error while preparing to receive incoming datagrams (query requests). The name server opens and configures a datagram socket for every local IP address. The name server could not set one of the sockets as reusable (the setsockopt() function failed).

System Action: The domain name server continues normal processing.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the setsockopt() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: opensocket

EZZ6496W error opening file *file name***:** *error information*

Explanation: The domain name server could not read the resource information from the indicated zone file. The fopen() or fstat() function failed.

System Action: The domain name server continues. The resource information in the indicated file is not loaded, and any queries concerning resources in the file will fail.

User or Operator Response: Verify the name of the file is correct and the file does exist and can be accessed by the domain name server. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the fopen() and fstat() functions and their possible error values.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6497W sysquery: no addresses found for a name server

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a name server. The name server being searched for was defined in the local name server's cache, was defined in one of its zone files, or was learned as part of the resource information obtained from another name server.

System Action: The domain name server continues normal processing. Problems may occur if the missing name server is needed to resolve other query requests or perform other name server operations.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6498E bind(dfd=*socket*, [*name*].*port*) failed: *error information*

Explanation: The domain name server encountered an error while preparing to receive incoming datagrams (query requests). The name server opens and configures a datagram socket for every local IP address. The name server could not set bind to one of the UDP sockets. (the bind() function failed).

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated. Ensure that the TCP/IP port definitions are correct. Refer to the *z/OS Communications Server: IP Configuration Reference* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the bind() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: opensocket

EZZ6502E can't open debug file *file name*: *error information*

Explanation: The domain name server encountered an error trying to open the file used to store debug information.

System Action: The domain name server ignores the debug option or request. It continues normal processing, but does not capture any debug information.

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

System Programmer Response: Verify the domain name server has write access to the directory and debug file and there is enough file space available to hold the debug information. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can restart the domain name server with the debug option, or send the -USR1 signal to the name server. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

See *IBM Open Class Library Reference, Vol. 1* for more information about the open() and fopen() functions and their possible error values.

Source Data Set: NS_MAIN

Procedure Name: setdebug

EZZ6504I query logging on

Explanation: The option to log all received resource queries is active.

System Action: The domain name server continues normal processing, logging each query request as it is received. The performance of the name server may be affected by this option, depending on the capacity of the system and the number of queries received by the name server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: setQrylogFlg

EZZ6505I query logging off

Explanation: The option to log all received resource queries is turned off.

System Action: The domain name server continues normal processing. The name server will no longer log each query request as it is received.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: setQrylogFlg

EZZ6507E sqadd: storage allocation failed: *error information*

Explanation: The domain name server encountered an error while trying to accept an incoming TCP connection. It did not have the storage needed to hold the information needed to use the connection.

System Action: The domain name server tries to recover and continue normal processing. The TCP connection is closed, but the remote host should recover. DNS resolvers will detect the connection failure and retry the connection or try and connect to a different name server.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (calloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the calloc() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: sqadd

EZZ6508I interface [*address***] missing; deleting**

Explanation: The indicated local IP address is no longer defined. This condition can occur when a local interface is removed from the system and the domain name server is re-cycled (reloaded).

System Action: The internal representation of the IP address is deleted. The datagram socket used to receive datagrams from the interface is closed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: dqflush

EZZ6509E getpeername failed: *error information*

Explanation: The domain name server could not obtain the host name of the name server it is initiating a zone transfer with. The getpeername() function failed.

System Action: The domain name server continues normal processing. The information was only used in debug and log messages.

User or Operator Response: No action is required. Inform the system programmer of the problem if the problem persists.

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the getpeername() function and its possible error values.

Source Data Set: NS_MAIN

Procedure Name: ns_setproctitle

EZZ6510I Received shutdown signal

Explanation: The domain name server was stopped. The name server was shutdown by an operator.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

User or Operator Response: Restart the domain name server when you want it to resume providing domain name services.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: nsexit

EZZ6511I Number of roots in zone too low (*number*)

Explanation: The domain name server is checkpointing its current zone data to a file. The domain name server determined the resulting checkpoint file may not be usable (it may not be a valid zone file that can be loaded back into the domain name server). The number of NS records (name server) with non-zero TTLs (time-to-live values) may not be sufficient.

System Action: The domain name server continues to create the checkpoint file. In addition to the zone information, all cache data is also placed in the checkpoint file. The assumption is this information may be useful in constructing the NS records needed to make a valid zone file. The beginning of the cache data in the file is indicated by the line ″; ---- Root hint cache dump ----″ and ended by the line ″; ---- Cache dump ----″. The information following that line is the zone information.

User or Operator Response: The resulting checkpoint file should not be reloaded into the domain name server as a zone file. To make a valid checkpoint file, edit the file and:

- Use the information contained in the cache data section of the file to determine the NS records that need to be inserted into the zone information section.
- Add the required NS records to the zone information section.
- Delete the cache data section (the ″Root hint″ section).

For more assistance contact the system programmer.

System Programmer Response: Assist the user if necessary. Determine why the zone file information does not contain enough NS records. Verify the TTL values on the NS records are valid and are compatible with the TTLs of zone's associated resource records.

Source Data Set: DB_DUMP

Procedure Name: doachkpt

EZZ6512I dumping name server resource data

Explanation: The domain name server is checkpointing its current zone data to a file.

System Action: The domain name server creates a checkpoint file containing all zone information contained in its internal databases. Temporary cache information is not included. The resulting checkpoint file can be used as a zone file to reload the domain name server with the checkpointed zone information.

The domain name server does not answer any resource queries until the checkpoint completes. Once the checkpoint is complete, normal processing resumes.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_DUMP

Procedure Name: doadump

EZZ6513I finished dumping name server resource data

Explanation: The domain name server finished checkpointing its current zone data. The resulting checkpoint file can be used as a zone file to reload the domain name server with the checkpointed zone information.

System Action: The domain name server resumes normal processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_DUMP

Procedure Name: doadump

EZZ6514C named: unrecoverable error: *error message*

Explanation: The domain name server encountered an unrecoverable error. The supplied *error message*, as well as any messages previously logged by the domain name server, identify the cause of the problem.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: DB_GLUE

Procedure Name: panic

EZZ6515C named: unrecoverable error: *error message*: *errno information*

Explanation: The domain name server encountered an unrecoverable error. The supplied error information, as well as any messages previously logged by the domain name server, identify the cause of the problem.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: The *errno information* was obtained using the strerror() function for the errno value associated with the failure. Use this information, along with information contained in any related messages, to determine the cause of the error. Use standard procedures to correct the error condition.

Source Data Set: DB_GLUE

Procedure Name: panic

EZZ6516E gettimeofday failed: *error information*

Explanation: The domain name server could not obtain the current time of day. The gettimeofday() function failed.

System Action: The domain name server continues, but unpredictable results may follow.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the gettimeofday() function and its possible error values.

Source Data Set: DB_GLUE

Procedure Name: gettime

EZZ6517W close(*socket***) failed:** *error information*

Explanation: The domain name server encountered an error attempting to close the indicated TCP/IP socket.

System Action: The domain name server continues. Problems may occur if the error re-occurs often, such as a shortage of available sockets for use by the domain name server.

User or Operator Response: Notify the system programmer if the error persists.

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the close() function and its possible error values.

Source Data Set: DB_GLUE

Procedure Name: my_close

EZZ6518W fclose(*file descriptor***) failed:** *error information*

Explanation: The domain name server encountered an error attempting to close a file.

System Action: The domain name server continues. Problems may occur if the error re-occurs often, such as a shortage of available file descriptors for use by the domain name server. In addition, the file might not be available for use by other programs.

User or Operator Response: Notify the system programmer if the error persists.

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the fclose() function and its possible error values.

Source Data Set: DB_GLUE

Procedure Name: my_fclose

EZZ6519E rm_name(*pointer***(***name***)): non-nil** *additional information* **pointer**

Explanation: The domain name server encountered an internal software error. A should-not-occur condition was detected.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages.

Source Data Set: DB_GLUE

Procedure Name: rm_name

EZZ6520W domain name too long: *name*

Explanation: The domain name server database contains an incorrectly formatted domain name. The length of the name exceeds the maximum allowed for domain names.

System Action: The domain name server continues. The string ″Name_Too_Long″ is used as the domain name. This name may be placed in a query reply, depending on the request being processed when the error occurred.

User or Operator Response: The invalid domain name may have been obtained from another domain name server or from one of the local domain name server zone files. Correct the local domain name server zone file information, if appropriate. Notify the system programmer if you suspect the invalid name was obtained from a remote domain name server.

System Programmer Response: Determine where the invalid domain name was obtained (using domain information contained in the name). Correct the zone file information for the domain name server in error.

Source Data Set: DB_GLUE

Procedure Name: getname

EZZ6522I *file name*: **line** *line number*: **number >** *max value*

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. A numeric value on that line exceeds the maximum allowed integer value.

System Action: The domain name server continues to load the zone information, substituting the maximum allowed value for the invalid value.

User or Operator Response: Fix the problem in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6523E *file name*: **Line** *line number*: **Unknown type:** *type*.

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The specified resource record type is invalid.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Update the resource information in the zone file to specify the correct type. Refer to *z/OS Communications Server: IP Configuration Reference* for the list of recognized types.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP User's Guide and Commands* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6524W *file name*: **line** *line number*: **ISDN-address too long**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The address information of an ISDN resource record is too long.

System Action: The domain name server truncates the address to 255 characters (the maximum allowed) and continues to load the zone information.

User or Operator Response: Fix the ISDN address information in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6525W *file name*: **line** *line number*: **CPU-type too long**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The CPU information associated with a HINFO resource record is too long.

System Action: The domain name server truncates the data to 255 characters (the maximum allowed) and continues to load the zone information.

User or Operator Response: Fix the HINFO CPU information in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6526W *file name*: **line** *line number*: **OS-type missing**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The HINFO resource record is incomplete. The OS-type (type of operating system) section of the record is missing.

System Action: The domain name server sets the OS-type to '?' and continues to load the zone information.

User or Operator Response: Add the type of operating system to the HINFO record in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6527W *file name*:*line number*: **ISDN-sa too long**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The sub-address information of an ISDN resource record is too long.

System Action: The domain name server truncates the sub-address to 255 characters (the maximum allowed) and continues to load the zone information.

User or Operator Response: Fix the ISDN sub-address information in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6528W *file name*:*line number*: **OS-type too long**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The OS-type (type of operating system) information associated with a HINFO resource record is too long.

System Action: The domain name server truncates the data to 255 characters (the maximum allowed) and continues to load the zone information.

User or Operator Response: Fix the operating system type information in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6529E *file name*:*line number*: **SOA-class not same as zone's**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The class specified on the SOA resource record does not match the class defined for the zone (in the domain name server boot file).

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Change the class information for the zone or the SOA record so they match. The class for SOA records is almost always IN (Internet). The other possible class type is HESIOD.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6530E *file name*: **line** *line number*: **SOA for** '*domain*' **not at zone top** '*origin*'

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The domain name specified on the SOA resource record does not match the domain origin information specified for the zone file in the domain name server boot file. This is a validation check to ensure that the files loaded by the domain name server are compatible.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the indicated zone. These names specify which domain name system domains (for example, ibm.com) the resources in the zone file are associated with.

User or Operator Response: Change the domain origin in the boot file or the domain name on the SOA record so they match.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6532E *file name*: **SOA increment time must be < 1/3 of the refresh time**

Explanation: The domain name server detected a syntax error in the SOA record within the indicated zone file. The increment serial number time period must be less than 1/3 of the zone refresh time period.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Change the SOA record so the refresh and increment serial number time periods are compatible.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6533W *file name*: **SOA expiration value is less than SOA refresh (***expiration value* < *refresh value***)**

Explanation: The domain name server detected a syntax error in the SOA record within the indicated zone file. The expiration time period is less than the zone refresh time period.

System Action: The domain name server uses the specified values and continues to load the zone information. The name server will not answer queries for the resources in the zone after the expiration time expires until the zone refresh time expires, at which time the zone information is refreshed.

User or Operator Response: Change the SOA record so the expiration time period is as long as the zone refresh time period.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6534I *file name*: **line** *line number*: **TXT record truncated**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The text information associated with a TXT resource record is too long.

System Action: The domain name server truncates the data to 2048 characters (the maximum allowed) and continues to load the zone information.

User or Operator Response: Fix the TXT record within the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6535E *file name*:*line number*: **data** '*domain*' **outside zone** '*origin*' **(ignored)**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The domain name specified on a resource record does not match the domain defined for the zone by the SOA record and the domain name server boot file. This is a validation check to ensure that the resources defined in the zone files are specified correctly.

System Action: The domain name server ignores the invalid resource record and continues to load the zone information.

User or Operator Response: Correct the domain name on the resource record or move the definition to the correct zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6536I *file name*: **line** *line number*: **database format error (***string***)**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The line does not contain a valid resource record. Either the format of the record or the format of the values within the record are invalid.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Fix the resource record definition on the indicated line within the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6537W ! UNSECURE MODE ! Dynamic Zone '*origin*' (file *file name*).

Explanation: The domain name server finished loading the resource data from the indicated zone file. The zone is defined as a zone capable of accepting dynamic resource updates, but the zone file did not contain any security information (KEY records).

System Action: The domain name server continues normal processing. Dynamic updates are allowed for the resources defined in the zone file, but no security checks are performed to validate the update requests.

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

System Programmer Response: Verify that you want to have non-verified resource update requests for this zone. If not, stop the domain name server, update the zone file to include DNS security information, and restart the domain name server. See *z/OS Communications Server: IP Configuration Guide* for more information about dynamic updating of resource information and the proper format of zone files.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6538W No active ZONE KEY RR defined. Dynamic updates disabled for Zone '*origin*' (file *file name*).

Explanation: The domain name server finished loading the resource data from the indicated zone file. The zone is defined as a zone capable of accepting dynamic resource updates, but the zone file did not contain any security information (KEY records).

System Action: The domain name server will respond to queries for the resources in the zone, but will not allow the resources in the zone to be dynamically updated. In order to allow dynamic updates for the resources in the zone, the zone file must contain DNS security information.

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

System Programmer Response: Change to the definition of the zone to prohibit dynamic updates or add the needed security information to the zone file.

After the problem is resolved, you can update the zone information in the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Guide* for more information about how to reload the zone information and dynamically updating resource information.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6539I *zone type* **zone '*origin*' rejected due to errors (serial *serial*)**

Explanation: The domain name server detected an error while loading the indicated zone information. Additional information about the types of errors encountered are contained in messages previously logged by the domain name server.

System Action: The resource information in the indicated zone is not loaded, and any queries concerning resources in the zone will fail.

User or Operator Response: Use the information contained in this and any previously logged messages to determine the cause of the error and respond as indicated.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6540I *zone type* **zone** '*origin*' **loaded (serial** *serial***)**

Explanation: The domain name server successfully loaded or updated the resource information for the indicated zone.

System Action: The domain name server resumes normal processing. It will respond to queries for the resources in the zone using the new information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6541I *file name*: **line** *line number*: **Unknown '$' option: $***option*

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The specified directive is not valid. Directives are statements that begin with '$', such as $ORIGIN or $INCLUDE.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the incorrect data.

User or Operator Response: Fix the syntax of the line in the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: gettoken

EZZ6542E *file name*:*line number*: **expected a number**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The line does not contain a valid resource record definition. Either the format of the record or the format of the values within the record are not valid.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Fix the resource record definition on the indicated line within the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: getnum

EZZ6543E *file name*:*line number*: **number after the decimal point exceeds 9999**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. A number specified on the line is formatted incorrectly. The domain name server only supports real numbers with up to four decimal places of precision (ten-thousandths).

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Fix the resource record definition on the indicated line within the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: getnum

EZZ6544I *file name*:*line number*: **decimal serial number interpreted as** *value*

Explanation: Informative message on how a numeric value is interpreted internally.

System Action: Normal processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_LOAD

Procedure Name: getnum

EZZ6545E *file name*: **line** *line number*: **unexpected end-of-file**

Explanation: The domain name server detected a syntax error in the zone file. One or more statements in the file were not completed.

System Action: The server does not use any of the resource information associated with the zone containing the invalid data.

User or Operator Response: Fix the syntax of the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: getnonblank

EZZ6546W *file name*: **line** *line number*: **unknown protocol:** *protocol*.

Explanation: The domain name server detected a problem or syntax error in the zone file at the indicated line number. The protocol specified by the WKS (well-known service) record on that line is not recognized by the domain name server. The specified protocol is either invalid or is not defined at the local system.

System Action: The domain name server continues to load the zone information, using -1 as the protocol number for the well-known service.

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

System Programmer Response: Fix the resource record definition on the indicated line within the zone file, add the definition of the missing protocol to the system, or remove the well-known service definition from the file. Refer to the *IBM Open Class Library Reference, Vol. 1* for the getprotoent() instruction, for information about the how the protocols are found on the local system.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Source Data Set: DB_LOAD

Procedure Name: getprotocol

EZZ6547W *file name*: **line** *line number*: **Unknown service** '*service*'

Explanation: The domain name server detected a problem or syntax error in the zone file at the indicated line number. The service specified by the WKS (well-known service) record on that line is not recognized by the domain name server. The specified service is either invalid or is not defined at the local system.

System Action: The domain name server continues to load the zone information, using -1 as the service number for the well-known service.

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

System Programmer Response: Fix the resource record definition on the indicated line within the zone file, add the definition of the missing service to the system, or remove the well-known service definition from the file. Refer to the *IBM Open Class Library Reference, Vol. 1* for the getserverices() instruction, for information about the how the services are found on the local system.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Source Data Set: DB_LOAD

Procedure Name: getservices

EZZ6548W *file name*: **line** *line number*: **port number (***port number***) too big**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The port number specified by the WKS (well-known service) record on that line is invalid (the value is too large).

System Action: The domain name server continues to load the zone information, truncating the port number.

User or Operator Response: Fix the resource record definition on the indicated line within the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: getservices

EZZ6549W *file name*: **line** *line number*: **missing closing parenthesis**

Explanation: The domain name server detected a syntax error in the zone file at the indicated line number. The format of the WKS (well-known service) definition on that line is invalid.

System Action: The domain name server continues to load the zone information. The definition of the well-known service is unpredictable.

User or Operator Response: Fix the WKS record definition on the indicated line within the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: getservices

EZZ6550W *statement* **contains invalid data (***element***)**

Explanation: The domain name server detected a syntax error in the boot file. The format of the indicated statement is invalid. It contains one or more invalid internet addresses (elements).

System Action: The domain name server continues to initialize, but the invalid internet address is ignored.

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

System Programmer Response: Fix the statement within the boot file. Restart the domain name server to pick up the changes. See *z/OS Communications Server: IP Configuration Reference* for more information about the proper format of the boot file.

Source Data Set: DB_LOAD

Procedure Name: get_netlist

EZZ6551I *string* **element** *element* **has bad mask (***mask***)**

Explanation: The domain name server detected a syntax error in the boot file. The format of the indicated statement is invalid. It contains one or more invalid subnet masks.

System Action: The domain name server continues to initialize, but the internet address associated with the invalid mask is ignored.

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

System Programmer Response: Fix the statement within the boot file. Restart the domain name server to pick up the changes. See *z/OS Communications Server: IP Configuration Reference* for more information about the proper format of the boot file.

Source Data Set: DB_LOAD

Procedure Name: get_netlist

EZZ6552I *string* **element (***element***) mask problem (***internet address***)**

Explanation: The domain name server detected a syntax error in the boot file. The contents of the indicated statement are invalid. The statement contains a list of internet addresses with an optional subnet mask for each address. One of the entries in the list (address and mask pair) is invalid because the specified internet address is not within the subnet defined by the subnet mask.

System Action: The domain name server continues to initialize, using the invalid entry.

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

System Programmer Response: Fix the statement within the boot file. Restart the domain name server to pick up the changes. See *z/OS Communications Server: IP Configuration Reference* for more information about the proper format of the boot file.

Source Data Set: DB_LOAD

Procedure Name: get_netlist

EZZ6553E **Zone '***origin***' (file** *file name***) no-SOA RR found**

Explanation: The domain name server detected a syntax error in the zone file. The zone file is missing the required SOA resource record.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the indicated zone.

User or Operator Response: Add the missing SOA record to the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6554E Zone '*origin*' (file *file name*) multiple SOA RRs found

Explanation: The domain name server detected a syntax error in the zone file. The zone file contains more than one SOA for the same zone.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the indicated zone.

User or Operator Response: /reRemove the extraneous SOA record definitions from the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6555E Zone '*origin*' (file *file name*) no NS RRs found at zone top

Explanation: The domain name server detected a syntax error in the zone file. The zone file does not contain a name server (NS) definition for the zone. The zone file must contain an NS definition specifying the host name of the name server for the zone.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server does not use any of the resource information associated with the indicated zone.

User or Operator Response: Add the missing NS record to the zone file.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6556W Zone '*origin*' (file *file name*) no-relevant RRs found

Explanation: The domain name server detected a possible problem in the zone file. The zone file is empty. It does not contain any resource definitions except for the SOA record for the zone and the NS record identifying the name server for the zone.

System Action: The domain name server continues to load the zone information, checking for additional errors. The server ignores the zone definition.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost. Having an empty zone file will not affect the processing of the rest of the zones, but removing it will simplify your configuration and possibly slightly improve the performance of the domain name server.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_LOAD

Procedure Name: db_load

EZZ6557I reloading name server resource information

Explanation: The domain name server is reloading its resource information from its zone files. The reload was triggered by a user request to refresh the name server's resource information.

System Action: The domain name server reloads all its resource information and the configuration data contained in the boot file. The name server does not answer any resource queries until its resource information is refreshed. Once the refresh is complete, normal processing resumes.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_RELOA

Procedure Name: db_reload

EZZ6558E named: savehash: storage allocation failed: *error information*

Explanation: The domain name server could not expand its internal cache. It did not have the storage needed to hold all the resource definitions and information known by the name server.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (malloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the malloc() function and its possible error values.

Source Data Set: DB_SAVE

Procedure Name: savehash

EZZ6559E build_secure_netlist (*origin*): storage allocation failed

Explanation: The domain name server could not expand its internal cache large enough to build the list of secure networks for a zone. It did not have the storage needed to hold all the resource definitions and information known by the name server.

System Action: The domain name server continues to load the zone information, using any secure network information obtained before the error occurred. The name server will allow these hosts to query and update the resources in the zone. Not all allowed hosts were loaded before the error occurred, so queries updates from some hosts that should be authorized will be rejected.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (malloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the malloc() function and its possible error values.

Source Data Set: DB_SECUR

Procedure Name: build_secure_netlist

EZZ6560W build_secure_netlist (*origin*): Bad address: *address*

Explanation: The domain name server detected a syntax error in the secure network information for a zone.

The secure network information for a zone is obtained from one or more TXT resource records within the zone with the resource name ″secure_zone″. For example, the secure network information for the zone ″ibm.com″ is contained in the TXT records within the zone file with the name ″secure_zone.ibm.com″.

The format of the TXT data is invalid. It contains one or more invalid internet addresses.

System Action: The domain name server continues to initialize, but the invalid internet address is ignored. The name server will not process queries or updates from that host for this zone.

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

System Programmer Response: Fix the definition of the secure network information for this zone.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information, the proper format of zone files, and how to define secure network information for a dynamic zone.

Source Data Set: DB_SECUR

Procedure Name: build_secure_netlist

EZZ6561W build_secure_netlist (*origin*): Bad mask: *mask*

Explanation: The domain name server detected a syntax error in the secure network information for a zone.

The secure network information for a zone is obtained from one or more TXT resource records within the zone with the resource name ″secure_zone″. For example, the secure network information for the zone ″ibm.com″ is contained in the TXT records within the zone file with the name ″secure_zone.ibm.com″.

The format of the TXT data is invalid. It contains one or more invalid subnet masks.

System Action: The domain name server continues to initialize, but the internet address associated with the invalid mask is ignored. The name server will not process queries or updates from that host for this zone.

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

System Programmer Response: Fix the definition of the secure network information for this zone.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information, the proper format of zone files, and how to define secure network information for a dynamic zone.

Source Data Set: DB_SECUR

Procedure Name: build_secure_netlist

EZZ6562W build_secure_netlist (*origin*): addr (*address*) is not in mask (*mask*)

Explanation: The domain name server detected a syntax error in the secure network information for a zone.

The secure network information for a zone is obtained from one or more TXT resource records within the zone with the resource name ″secure_zone″. For example, the secure network information for the zone ″ibm.com″ is contained in the TXT records within the zone file with the name ″secure_zone.ibm.com″.

The format of the TXT data is invalid. The data contains a list of internet addresses with an optional subnet mask for each address. One of the entries in the list (address and mask pair) is invalid because the specified internet address is not within the subnet defined by the subnet mask.

System Action: The domain name server continues to initialize, using the invalid entry. The name server may not allow queries or updates from that host for this zone.

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

System Programmer Response: Fix the definition of the secure network information for this zone.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information, the proper format of zone files, and how to define secure network information for a dynamic zone.

Source Data Set: DB_SECUR

Procedure Name: build_secure_netlist

EZZ6563W build_secure_netlist (*origin*): duplicate address *address*

Explanation: The domain name server detected a syntax error in the secure network information for a zone.

The secure network information for a zone is obtained from one or more TXT resource records within the zone with the resource name ″secure_zone″. For example, the secure network information for the zone ″ibm.com″ is contained in the TXT records within the zone file with the name ″secure_zone.ibm.com″.

The format of the TXT data is invalid. It contains more than one entry for the same internet addresses.

System Action: The domain name server continues to initialize, but the duplicate definitions are ignored.

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

System Programmer Response: Fix the definition of the secure network information for this zone. Remove the duplicate definitions, or change the definition to specify the correct address.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information, the proper format of zone files, and how to define secure network information for a dynamic zone.

Source Data Set: DB_SECUR

Procedure Name: build_secure_netlist

EZZ6564I *name* has CNAME and other illegal data

Explanation: The domain name server encountered a problem while trying to update the definition of a resource. The name server detected that other resource records in the zone are using the same name as a CNAME record within the zone.

System Action: The domain name server ignores the requested update and continues normal processing.

User or Operator Response: Correct the resource definitions in the zone so the names of the CNAME resource records do not conflict with any other resource definitions.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary.

Source Data Set: DB_UPDAT

Procedure Name: db_update

EZZ6565I domain *name type* record in zone *new zone* should be in zone *old zone*, ignored

Explanation: The domain name server received an update for a resource which indicates it is in a different zone (*new zone*) than the zone indicated by the existing definition of the resource (*old zone*).

System Action: The credibility level of the existing definition is higher, so the received update is ignored. The resource remains defined in its current zone.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_UPDAT

Procedure Name: db_update

EZZ6566I domain *name type* **record in zone** *old zone* **should be in zone** *new zone***, deleted**

Explanation: The domain name server received an update for a resource which indicates it is in a different zone (*new zone*) than the zone indicated by the existing definition of the resource (*old zone*).

System Action: The credibility level of the received update is higher, so the existing definition of the resource within the old zone is removed. The resource should be added to the new zone as part of the normal processing of dynamic updates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DB_UPDAT

Procedure Name: db_update

EZZ6567E named-xfer: can not make temporary file (*file name***):** *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The temporary file used to hold the zone data could not be created.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the domain name server has access to the directory indicated in the message, it can create temporary files within the directory, and there is enough file space available to hold the temporary file. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the open() function and its possible error values.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6568E named-xfer: can not [f]chmod temporary file (*file name***):** *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The required access controls could not be set for the file. The name server attempts to access the file as read/write and set read permission for other users (permission bits 0644).

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the domain name server has the correct access to the directory and files and there is enough file space available to hold the temporary file. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the chmod() and fchmod() functions and their possible error values.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6569E named-xfer: can not fdopen temporary (*file name*)

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not open the temporary file being used to receive zone data.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the domain name server has the correct access to the directory and files and there is enough file space available to hold the temporary file. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the open() and fdopen() functions and their possible error values.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6570I unknown server (*name*) for *origin*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not determine the IP address of the indicated name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the zone information is defined correctly. The secondary zone definitions should contain the correct TCP/IP host name for the primary name server, and usually contain the IP address of the name server. Check the SOA and NS records in the zone file.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

If you cannot determine the cause of the problem, contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. In addition include copies of the zone files for the name servers involved in the problem. If possible, dump the domain name server's resource data and statistics and provide that information as well. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data and use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6571I SOA query address = 0.0.0.0 (*name*) for *origin*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not determine the IP address of the indicated name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource

information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the zone information is defined correctly. The secondary zone definitions should contain the correct TCP/IP host name for the primary name server, and usually contain the IP address of the name server. Check the SOA and NS records in the zone file.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis*for more information.

If you cannot determine the cause of the problem, contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. In addition include copies of the zone files for the name servers involved in the problem. If possible, dump the domain name server's resource data and statistics and provide that information as well. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data and use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6572E named-xfer: unable to rename *file name* **to** *zone file name***:** *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone information was successfully received, but the temporary file could not be renamed or copied to the actual zone file.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify the domain name server has access to the directory indicated in the message, it can create temporary files within the directory, it can write to and rename the actual zone file, and there is enough file space available. For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the rename() function and its possible error values.

Source Data Set: NAMED@XF

Procedure Name: main

EZZ6573E storage allocation (*size***) failed**

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not obtain the storage required to receive the transfer.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6574E socket failed for a zone transfer: *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not create a socket for the TCP connection to the primary name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the socket() function and its possible error values.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6575E connect(*address***) for zone** *origin* **failed:** *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not connect to the primary name server (using the IP address shown in the message).

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the connect() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6576E zone *origin***: res_mkquery T_SOA failed**

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred trying to create and send a query for the SOA record to the primary name server. Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6577E send query failed for a zone transfer: *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred while trying to send a query request to the primary name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the writev() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6578I storage allocation (*size*) failed for SOA from server [*address*], zone *origin*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server could not obtain the storage required to receive the transfer.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server.

See *IBM Open Class Library Reference, Vol. 1* for more information about the realloc() and malloc() functions and their possible error values.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6579E bad response to SOA query from [*address*], zone *origin***: rcode** *request***, aa** *value***, ancount** *answer count***, aucount** *authority count*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received an invalid response to its query for the SOA record from the indicated primary name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the

resulting debug file. See *z/OS Communications Server: IP User's Guide and Commands* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6580E malformed SOA from [*address***], zone** *origin***:** *message*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received invalid SOA information from the indicated primary name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6581E wrong query in response from [*address***], zone** *origin***: [***name class type***]**

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received an invalid response to its query for the SOA record from the indicated primary name server. It received a response for an unrelated query.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP User's Guide and Commands* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6582E wrong answer in response from [*address***], zone** *origin***: [***name class type***]**

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received an invalid response to its query for the SOA record from the indicated primary name server. It received a response for an unrelated query.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6583E zone *origin*: res_mkquery T_NS failed

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred trying to create and send a query for the NS record to the primary stub name server. The primary stub name server is a proxy name server, and the NS record contains the address information for the real primary name server. Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6584E zone *origin*: res_mkquery T_AXFR failed

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred trying to create and send an AXFR query to the primary name server. The AXFR query starts the actual zone transfer.

Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6585E record too short from [*address*], zone *origin*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received an invalid query response from the indicated primary name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6586E print_output: short answer (*response size*, *record size*), zone *origin*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received invalid or unknown resource information from the indicated primary name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Either the name servers are not compatible or one contains a software error. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6587E wrong zone name in AXFR (wanted '*name1*', got '*name2*')

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received invalid data in the response to the AXFR or SOA query from the primary name server. The zone name in the response is incorrect.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6588E received extra SOA for '*name*' in zone '*zone name*'

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received invalid data in the response to the AXFR or SOA query from the primary name server. The response contains multiple SOA records, or contained an SOA record for another zone.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6589I serial from [*address*], zone *origin*: *new serial number* lower than current: *serial number*

Explanation: The domain name server encountered an unexpected condition while processing a zone transfer. The name server is a secondary or caching name server for the zone. The primary name server's serial number for the zone is less than the secondary name server's value for the same zone. This indicates that the zone data at the secondary name server is more up to date then the data at the primary name server, which should not occur.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: This condition may be caused by the administrator at the primary name server updating the serial number for the zone to a lesser value. The primary name server should have refreshed the secondary name servers when this occurred, but an error may have occurred. Restart the secondary name server.

If the problem persists, Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: getzone

EZZ6590E setitimer failed: *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred processing the response to a query from the primary name server. The setitimer() function failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the setitimer() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NAMED@XF

Procedure Name: netread

EZZ6591E premature end-of-file reached, fetching '*domain*'

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred trying to receive the response to a query from the primary name server. No data was received.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

Source Data Set: NAMED@XF

Procedure Name: netread

EZZ6592E recv(len=*length*) failed: n=*length* but no error information

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred trying to receive the response to a query from the primary name server. The recv() function failed but there isn't any other error information available.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: See *IBM Open Class Library Reference, Vol. 1* for more information about the recv() function and its possible error values. The error was likely caused by a network failure.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

Source Data Set: NAMED@XF

Procedure Name: netread

EZZ6593E recv(len=*length*) failed: *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. A failure occurred trying to receive the response to a query from the primary name server. The recv() function failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP User's Guide and Commands* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the recv() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NAMED@XF

Procedure Name: netread

EZZ6594E '*name class type*' **- unknown type (***value***)**

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The name server received invalid or unknown resource information from the indicated primary name server. The domain name server does not understand one or more of the resource types contained in the received resource information.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Either the name servers are not compatible or one contains a software error. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS*

Communications Server: IP Configuration Referencez/OS Communications Server: IP User's Guide and Commands for information about how to use its debugging options.

Source Data Set: NAMED@XF

Procedure Name: print_output

EZZ6595E increment_serial: lookup failed on zone (*zone name*)

Explanation: An internal error occurred while processing an update request. The domain name server could not locate the SOA record for the internal definition of the zone.

System Action: The received update request is ignored. The domain name server continues normal processing. The behavior may be unpredictable, since it appears some of the internal data areas are corrupted.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message. If possible, dump the domain name server's resource data and statistics. See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data.

Source Data Set: NS_DYNDN

Procedure Name: increment_serial

EZZ6596E increment_serial: Couldn't find SOA record for '*zone name*'

Explanation: An internal error occurred while processing an update request. The domain name server could not locate the SOA record for the internal definition of the zone.

System Action: The received update request is ignored. The domain name server continues normal processing. The behavior may be unpredictable, since it appears some of the internal data areas are corrupted.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message. If possible, dump the domain name server's resource data and statistics. See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data.

Source Data Set: NS_DYNDN

Procedure Name: increment_serial

EZZ6597E Zone '*zone name*' declared more than once

Explanation: An internal error occurred while trying to recover from a resource update failure. When an error occurs processing a resource update, the domain name server tries to recover by restoring the original resource information. The name server encountered two internal representations of the same zone information, which should not occur.

System Action: The recovery process continues, restoring the information to the most likely of the two representations. The domain name server continues normal processing. The behavior may be unpredictable, since it appears some of the internal data areas are corrupted.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message. If possible, dump the domain name server's resource data and statistics. See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data.

Source Data Set: NS_DYNDN

Procedure Name: restore_one_zone

EZZ6598I forw_update: sendto(*socket address*) failed: *error information*

Explanation: The domain name server encountered an error while attempting to forward an update request for a resource to the name server that is authoritative for the resource. The TCP/IP sendto() function failed, preventing the transmission of the update.

System Action: The domain name server continues normal processing. The updated resource may be improperly defined in the domain name system, since the authoritative name server for the resource did not receive the update.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the sendto() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NS_DYNDN

Procedure Name: forw_update

EZZ6599E forw: storage allocation failed: *error information*

Explanation: The domain name server could not forward a query request to other name servers for resolution. The name server is trying to forward the request to other name servers because it did not have the information needed to answer the query. It could not forward the request because it did not have the storage needed to build and send the request.

System Action: The domain name server sends back a response indicating an error occurred while processing the query. The name server continues normal processing.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: No immediate action is required unless the error persists. The next time you start the domain name server, increase the amount of storage available for it. The *error information* in this message provides additional information about the cause of the memory allocation failure (malloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the malloc() function and its possible error values.

Source Data Set: NS_FORW

Procedure Name: ns_forw

EZZ6600W ns_forw: sendto([*address***].***port***) failed:** *error information*

Explanation: The domain name server could not forward a query request to another name server for resolution. The name server is trying to forward the request to another name server because it did not have the information needed to answer the query. The TCP/IP sendto() function failed, preventing the forwarding of the request to the other name servers.

System Action: The domain name server continues normal processing. The name server will retry forwarding the request after its retry time-out expires, as well as trying to forward the request to other name servers.

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

System Programmer Response: No immediate action is required unless the error persists. Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the sendto() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NS_FORW

Procedure Name: ns_forw

EZZ6601W Bogus (0.0.0.0) A RR for *name*

Explanation: The domain name server detected a possible problem in the name server's configuration information. The indicated name server is defined (there exists an NS record) but there is no corresponding definition for the address of the name server (for example, no A record). The invalid name server information could have been learned from another name server, but the more probable error is the local zone files are not correct.

System Action: The domain name server ignores the name server definition and continues normal processing.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Reference* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the local zone information appears to be correct, try to determine the name server that provided the invalid definition and fix its configuration. No action is required if the default network mask is acceptable.

Source Data Set: NS_FORW

Procedure Name: nslookup

EZZ6602I Bogus LOOPBACK A RR for *name*

Explanation: The domain name server detected a possible problem in the name server's configuration information. The indicated name server is defined (there exists an NS record) but the address defined for the name server is the TCP/IP loopback address. The invalid name server information could have been learned from another name server, but the more probable error is the local zone files are not correct.

System Action: The domain name server ignores the name server definition and continues normal processing.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the local zone information appears to be correct, try to determine the name server that provided the invalid definition and fix its configuration.

Source Data Set: NS_FORW

Procedure Name: nslookup

EZZ6603W Bogus BROADCAST A RR for *name*

Explanation: The domain name server detected a possible problem in the name server's configuration information. The indicated name server is defined (there exists an NS record) but the address defined for the name server is the TCP/IP broadcast address (255.255.255.255). The invalid name server information could have been learned from another name server, but the more probable error is the local zone files are not correct.

System Action: The domain name server ignores the name server definition and continues normal processing.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the local zone information appears to be correct, try to determine the name server that provided the invalid definition and fix its configuration.

Source Data Set: NAMED@XF

Procedure Name: nslookup

EZZ6604W Bogus MULTICAST A RR for *name*

Explanation: The domain name server detected a possible problem in the name server's configuration information. The indicated name server is defined (there exists an NS record) but the address defined for the name server is a TCP/IP multi-cast address. The invalid name server information could have been learned from another name server, but the more probable error is the local zone files are not correct.

System Action: The domain name server ignores the name server definition and continues normal processing.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the local zone information appears to be correct, try to determine the name server that provided the invalid definition and fix its configuration.

Source Data Set: NS_FORW

Procedure Name: nslookup

EZZ6605E named: forw: storage allocation failed: *error information*

Explanation: The domain name server could not expand its internal cache. It did not have the storage needed to hold all the resource definitions and information known by the name server.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Increase the amount of storage available for the domain name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (calloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the calloc() function and its possible error values.

Source Data Set: NS_FORW

Procedure Name: qnew_tagged, qnew

EZZ6606E named: not enough memory to allocate initial zones array

Explanation: The domain name server did not have the storage needed to hold all the resource definitions and information defined to the name server.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (malloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the malloc() function and its possible error values.

Source Data Set: NS_INIT

Procedure Name: ns_init

EZZ6607I Zone '*name*' was removed

Explanation: The domain name server reloaded its resource information from the zone files. The indicated zone is no longer defined.

System Action: All resources associated with the zone are removed. The domain name server continues normal processing, but will no longer respond to queries for resources in the purged zone.

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

System Programmer Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

Source Data Set: NS_INIT

Procedure Name: ns_init

EZZ6608E named: cannot open boot file '*file name*': *error information*

Explanation: The domain name server could not read its initial configuration file.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Verify that the format and contents of the boot file and all associated zone files are correct and can be read by the domain name server. Restore the configuration files from backup copies if the files have inadvertently been lost or corrupted. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated. See *IBM Open Class Library Reference, Vol. 1* for more information about the open() and fopen() functions and their possible error values.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6609C named: could not access directory *directory name*: *error information*

Explanation: The domain name server could not access the directory specified by the "directory" keyword in the boot file.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Verify that the format and contents of the boot file are correct, the indicated directory exists, the zone files assumed to be in the directory exist, and all files and directories can be read by the domain name server. Restore the configuration files from backup copies if the files have inadvertently been lost or corrupted. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

For other errors, use the *error information* in this message to determine the cause of this error and respond as indicated. See *IBM Open Class Library Reference, Vol. 1* for more information about the chdir() function and its possible error values.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6610I cache directive with non-IN class is not supported

Explanation: The domain name server detected a syntax error in the boot file. The resource class defined for a cache defined in the domain name server boot file is incorrect. The only resource class allowed is IN (Internet).

System Action: The domain name server ignores the cache keyword and continues normal processing. The resources defined in the zone file associated with the keyword are not loaded and the name server will not respond to queries for those resources.

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

System Programmer Response: Change the definition of the cache to have the correct resource class and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6611I *file name*: **line** *line number*: **unknown directive** '*directive*'

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The line contains an invalid keyword.

System Action: The domain name server ignores the line and continues normal processing.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6612I *file name*: **line** *line number*: **missing origin**

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The zone definition on the indicated line does not identify the zone name. The primary, secondary, stub, and cache keywords are used to define zone information and require the name of the zone on the definition.

System Action: The domain name server ignores the zone definition and continues normal processing. The resources defined in the associated zone file are not loaded and the name server will not respond to queries for those resources.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6613I *file name*: **line** *line number*: **zone** '*name*' **has trailing dot**

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The zone name on the indicated line ends in one or more dots ('.').

System Action: The domain name server strips the trailing dots from the name and continues normal processing. The resources defined in the associated zone file are loaded.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6614I *file name*: **line** *line number*: **missing** *missing item*

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The zone definition on the indicated line is incorrect. It is missing the indicated information.

System Action: The domain name server ignores the zone definition and continues normal processing. The resources defined in the associated zone file are not loaded and the name server will not respond to queries for those resources.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6615C no memory for more zones

Explanation: The domain name server did not have the storage needed to hold all the resource definitions and information defined to the name server.

System Action:

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6616I *file name*: **line** *line number*: **cache refresh ignored**

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The cache definition on that line contains a refresh time, but cache data is not automatically refreshed.

System Action: The domain name server ignores the refresh value and continues normal processing. The resources defined in the associated zone file are loaded.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6617I *file name*: **line** *line number*: **bad flag** '*flag*'

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The zone definition on that line contains an invalid or extraneous keyword option.

System Action: The domain name server ignores the extraneous option and continues normal processing. The resources defined in the associated zone file are loaded.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6618W *file name*: **line** *line number*: **bad security keyword** '*keyword*'

Explanation: The domain name server detected a syntax error in the boot file at the indicated line number. The zone definition on that line contains an invalid or extraneous keyword option.

System Action: The domain name server ignores the extraneous option and continues normal processing. The resources defined in the associated zone file are loaded.

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

System Programmer Response: Fix the line in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6619I **No reload of zone** '*origin*': **dynamic zone**

Explanation: The domain name server is in the process of reloading its resource information from the zone files. The identified zone is defined as a zone that supports dynamic updates. Zones that can be dynamically updated are not reloaded because the name server may have learned of resource changes that are not reflected in the zone file.

System Action: The domain name server continues normal processing, reloading resource information for all zones that do not support dynamic updates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6620W **Forwarding address** '*string*' **ignored, not a proper IP address.**

Explanation: The domain name server detected a syntax error in the boot file. The indicated data associated with the ″forwarders″ keyword is invalid. The proper value of the forwarder keyword is a list of IP addresses.

System Action: The domain name server ignores the invalid portion of the forwarder definition and continues normal processing. All other valid forwarder addresses are used.

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

System Programmer Response: Fix the forwarder definition in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: get_forwarders

EZZ6621I **forwarder** '*address*' **ignored, (local address)**

Explanation: The domain name server detected a syntax error in the boot file. The list of IP addresses associated with the ″forwarders″ keyword contains the local system's IP address.

System Action: The domain name server ignores the indicated address in the forwarder definition and continues normal processing. All other valid forwarder addresses are used.

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

System Programmer Response: Fix the forwarder definition in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: get_forwarders

EZZ6622C named: unrecognized limit in boot file: '*name*'

Explanation: The domain name server detected a syntax error in the boot file. The file contains an invalid value for the "limit" keyword.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Fix the limit definitions in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: ns_limit

EZZ6623W limit in boot file: '*name*' not implemented

Explanation: The domain name server detected a syntax error in the boot file. The file contains an unrecognized or unsupported value for the "limit" keyword.

System Action: The domain name server ignores the limit and continues normal processing.

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

System Programmer Response: Fix the limit definition in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: ns_rlimit

EZZ6624W getrlimit(*name*): *error information*

Explanation:

System Action:

User or Operator Response:

System Programmer Response:

Source Data Set: NS_INIT

Procedure Name: ns_rlimit

EZZ6625W setrlimit(*name*, *value*): *error information*

Explanation:

System Action:

User or Operator Response:

System Programmer Response:

Source Data Set: NS_INIT

Procedure Name: ns_rlimit

EZZ6626E named: unrecognized option in boot file: '*name*'

Explanation: The domain name server detected a syntax error in the boot file. The file contains an invalid value for the "options" keyword.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Fix the option definitions in the boot file and restart the domain name server. See *z/OS Communications Server: IP Configuration Reference* for a description of the configuration files and their format.

Source Data Set: NS_INIT

Procedure Name: ns_option

EZZ6627I qserial_query(*origin*): sysquery failed

Explanation: The domain name server encountered an error while attempting to determine if a zone transfer is required. The name server could not obtain the serial number of the zone from the primary name server for the zone.

System Action: The domain name server will attempt to determine if a transfer is needed at the next scheduled refresh period. The resource information for the zone is not refreshed.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_MAINT

Procedure Name: qserial_query

EZZ6628I Err/TO: error getting serial number for '*origin*'

Explanation: The domain name server encountered an error while attempting to determine if a zone transfer is required. The name server could not obtain the serial number of the zone.

System Action: The domain name server will attempt to determine if a transfer is needed at the next scheduled refresh period. The resource information for the zone is not refreshed.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_MAINT

Procedure Name: qserial_answer

EZZ6629I Zone '*origin*' (class *class*) SOA serial# (*primary serial number*) received from [*address*] is < ours (*serial number*)

Explanation: The domain name server encountered an unexpected condition while attempting to determine if a zone transfer is required. The name server is a secondary or caching name server for the zone. The primary name server's serial number for the zone is less than the name server's value for the same zone. This indicates that the zone data at the secondary name server is more up to date then the data at the primary name server, which should not occur.

System Action: The domain name server will attempt to determine if a transfer is needed at the next scheduled refresh period. The resource information for the zone is not refreshed.

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

System Programmer Response: This condition may be caused by the administrator at the primary name server updating the serial number for the zone to a lesser value. The primary name server should have refreshed the secondary name servers when this occurred, but an error may have occurred. Restart the secondary name server.

If the problem persists, contact the IBM Software Support Center or the vendor that supplied the primary domain name server. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP User's Guide and Commands* for information about how to use its debugging options.

Source Data Set: NS_MAINT

Procedure Name: qserial_answer

EZZ6630C attempted to fetch zone *origin* **from self (***address***)**

Explanation: An internal error occurred within the domain name server. The name server attempted to initiate a zone transfer with itself. The name server determined that its internal data areas are corrupted.

System Action: The domain name server continues normal processing. The behavior may be unpredictable, since it appears some of the internal data areas are corrupted.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message. If possible, dump the domain name server's resource data and statistics, and then restart the name server. See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data and how to control the name server.

Source Data Set: NS_MAINT

Procedure Name: startxfer

EZZ6631E named-xfer vfork failed: *error information*

Explanation: The domain name server encountered an error while attempting to initiate a zone transfer. The name server is a secondary or caching name server for the zone. The name server process that receives the zone information (named-xfer) could not be started. The vfork() function failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the vfork() function and its possible error values.

Source Data Set: NS_MAINT

Procedure Name: startxfer

EZZ6632E named: can't exec *path***:** *error information*

Explanation: The domain name server encountered an error while attempting to initiate a zone transfer. The name server is a secondary or caching name server for the zone. The name server process that receives the zone information (named-xfer) could not be started correctly. The execv() function failed for the named-xfer process.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the execv() function and its possible error values.

Source Data Set: NS_MAINT

Procedure Name: startxfer

EZZ6633W zone transfer time-out for '*origin*'; pid *process ID* missing

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out. The process ID of the zone transfer process (named-xfer) is unknown.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the time-out and respond as indicated. The error was likely caused by a network failure or poor network throughput and a large zone transfer.

The domain name server could not determine the ID of the zone transfer process, so it could not stop it. Determine if the process is still active, and if so, stop it.

Attention: There may be other valid zone transfer processes active (for transfers of other zones). These processes all look the same to the system (the only difference is the parameters used to start them). Make sure you do not stop a transfer process for another zone.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

Source Data Set: NS_MAINT

Procedure Name: abortxfer

EZZ6634W zone transfer timeout for '*origin*'; kill pid *process ID* failed: *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out and the request to stop the zone transfer process (named-xfer) failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

The domain name server will attempt to stop the zone transfer process one more time (after a delay).

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the time-out and respond as indicated. The error was likely caused by a network failure or poor network throughput and a large zone transfer.

The domain name server tried to stop the zone transfer process, but the kill() function failed. Determine if the indicated process is still active, and if so, stop it.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the kill() function and its possible error values.

Source Data Set: NS_MAINT

Procedure Name: abortxfer

EZZ6635W zone transfer timeout for '*origin*'; second kill pid *process ID* failed

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out and the second attempt to stop the zone transfer process (named-xfer) failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource

information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

The domain name server will no longer try to stop the zone transfer process.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the time-out and respond as indicated. The error was likely caused by a network failure or poor network throughput and a large zone transfer.

The domain name server tried to stop the zone transfer process, but the kill() function failed. Determine if the indicated process is still active, and if so, stop it.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the kill() function and its possible error values.

Source Data Set: NS_MAINT

Procedure Name: abortxfer

EZZ6636W zone transfer timeout for '*origin***'; pid** *process ID* **kill failed** *error information*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out and the request to stop the zone transfer process (named-xfer) failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

The domain name server will attempt to stop the zone transfer process one more time (after a delay).

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the time-out and respond as indicated. The error was likely caused by a network failure or poor network throughput and a large zone transfer.

The domain name server tried to stop the zone transfer process, but the kill() function failed. Determine if the indicated process is still active, and if so, stop it.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the kill() function and its possible error values.

Source Data Set: NS_MAINT

Procedure Name: abortxfer

EZZ6637W zone transfer timeout for '*origin***'; pid** *process ID* **killed**

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out and the process was terminated.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the time-out and respond as indicated. The error was likely caused by a network failure or poor network throughput and a large zone transfer.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

Source Data Set: NS_MAINT

Procedure Name: sched_maint

EZZ6638I zone transfer timeout for '*origin*'; pid *process ID* killed

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out. The zone transfer process was terminated.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the time-out and respond as indicated. The error was likely caused by a network failure or poor network throughput and a large zone transfer.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

Source Data Set: NS_MAINT

Procedure Name: sched_maint

EZZ6638N named-xfer ended with signal *signal*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. Information contained in messages previously logged by the domain name server may help identify the cause of the problem. The process used to receive the transferred data ended with the indicated signal value.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_MAINT

Procedure Name: endxfer

EZZ6639I zoneref: primaries for secondary zone '*origin*' unreachable

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. The zone transfer timed out.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Verify that the format and contents of your zone files are correct. Check to make sure the IP address of the primary name servers for the indicated zone are correct and the name servers are active. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

The error could also be caused by network failures or poor network throughput and a large zone transfer.

Source Data Set: NS_MAINT

Procedure Name: endxfer

EZZ6640I named-xfer for '*origin*' **ended** *value*

Explanation: The domain name server encountered an error while processing a zone transfer. The name server is a secondary or caching name server for the zone. Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The resource information for the zone is not refreshed, and any queries concerning resources in the zone will fail until the zone is successfully transferred.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

If you cannot determine the cause of the problem or the problem persists, contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. In addition include copies of the zone files for the name servers involved in the problem. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NS_MAINT

Procedure Name: endxfer

EZZ6641I zones changed: *old zone* **!=** *new zone*

Explanation: An unexpected condition occurred within the domain name server. The name server determined that some of its internal data areas are out-of-sync. It recovered from the problem.

System Action: The domain name server continues normal processing. Although it did recover, unexpected errors may still occur because there could be a software logic problem within the name server.

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

System Programmer Response: Contact the IBM Software Support Center if the problem reoccurs or the name server encounters unexplained problems. Dump the domain name server's resource data and statistics, and then restart the name server. Finally, if the problem occurs often or can be re-created, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to start the name server and use its debugging options and how to request a dump of the name server's resource data.

Source Data Set: NS_MAINT

Procedure Name: tryxfer

EZZ6642I zones shrunk

Explanation: An unexpected condition occurred within the domain name server. The name server determined that some of its internal data areas are out-of-sync. It recovered from the problem.

System Action: The domain name server continues normal processing. Although it did recover, unexpected errors may still occur because there could be a software logic problem within the name server.

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

System Programmer Response: Contact the IBM Software Support Center if the problem re-occurs or the name server encounters unexplained problems. Dump the domain name server's resource data and statistics, and then restart the name server. Finally, if the problem occurs often or can be re-created, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to start the name server and use its debugging options and how to request a dump of the name server's resource data.

Source Data Set: NS_MAINT

Procedure Name: tryxfer

EZZ6643E ns_req: sendto(*name*) failed: *error information*

Explanation: A failure occurred while sending a query response to the indicated TCP/IP host. The TCP/IP sendto() function failed, preventing the transmission of the response.

System Action: The domain name server continues normal processing. The response is lost but the remote host should retry the query. DNS resolvers will time out waiting for the reply and resend the query.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the sendto() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NS_REQ

Procedure Name: ns_req

EZZ6645W WLM load failed for *name* **db_update rc=** *value*

Explanation: The domain name server encountered an internal error while trying to add Work Load Manager load balancing information to the resource database.

System Action: The domain name server continues normal processing. The address of the server application can be queried, but the replies will be answered without considering the current capacity information. The name server will retry obtaining the data when the WLM refresh timer expires.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

The possible values are:

- -1 = NONAME - No name specified for host
- -4 = NODATA - No IP addresses specified
- -5 = DATAEXISTS - The name specified already exists
- -10 = AUTH - not authorized

Source Data Set: WLM_LOAD

Procedure Name: update_db

EZZ6646W WLM data error in group *group*

Explanation: The IP address information obtained for one or more server applications in the indicated group is invalid.

System Action: The domain name server continues normal processing.

If the server application is defined in the zone file, then the IP address of the server application can be queried, but the replies will be answered without considering current capacity.

If the server application is not defined in the zone, the IP address of the server cannot be determined from the name server. Any remote systems that wish to use the server must obtain the address via some other means.

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

System Programmer Response: Verify that the server application is registering its TCP/IP address information correctly. After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Guide* for more information about how to define server applications to the name server and how to register these applications with WLM.

Source Data Set: WLM_LOAD

Procedure Name: wlm_load

EZZ6647W Zone '*zone name*' WLM detected error, reason code = *value*

Explanation: The domain name server encountered an error while trying to obtain the list of TCP/IP server applications using the Work Load Management (WLM) facility for this zone. One of the issues WLM functions failed.

System Action: The domain name server continues normal processing. Connection balancing will not occur in this zone and the cluster names will not be defined in this zone. The name server will retry obtaining the data when the WLM refresh timer expires.

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

System Programmer Response: Use the reason code supplied by this message to determine the cause of this error. Respond as indicated. See *z/OS MVS Programming: Workload Management Services* for more information about the WLM functions (especially IWMDNGRP and IWMDNSRV) and their error values.

Source Data Set: WLM_LOAD

Procedure Name: wlm_load

EZZ6649E named: WLM group '*group name*' not created

Explanation: The domain name server encountered an internal error while trying to add the server applications for the indicated group to the name server's database.

System Action: The domain name server continues normal processing, but none of the server applications are registered with their load balancing information to the name server. Any resources previously defined in the zone can be queried, but the replies will be answered without considering current capacity. The name server will retry obtaining the data when the WLM refresh timer expires.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated. A named -d11 trace will give more information for the reason for this failure. If the trace shows that the WLM group name is already in use, ensure that there is not an A record for the WLM group name and that the WLM group name is not the high-level qualifier of the zone name in the SOA record. If the trace shows that the WLM group name could not be updated, ensure that the IP addresses that the stack advertises intersects with the IP addresses in the A records for the hostname in the cluster zone file. For the stack to advertise IP addresses to the name server, the stack must register with WLM using the IPCONFIG SYSPLEXROUTING statement, and the adapter associated with the IP address must be active.

Source Data Set: WLM_LOAD

Procedure Name: wlm_load

EZZ6650W rejected UDP AXFR from *hostname* for '*domainname*'

Explanation: The domain name server received an invalid request. The name server received a request to initiate a zone transfer via one of its datagram (UDP) sockets. Zone transfers are only performed using TCP connections.

System Action: The domain name server rejects the received request and continues normal processing.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the name server that sent the request. Supply the information contained in the message and any other related domain name server messages. Finally, if possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6651I unapproved AXFR from *hostname* **for** *zone*

Explanation: The domain name server received a zone transfer request from an unapproved secondary name server. The secondary name server is not in the list of name servers allowed to receive zone transfers from this name server. The list of approved name servers is specified using the ″xfrnets″ keyword in the domain name server boot file.

System Action: The domain name server rejects the received request and continues normal processing.

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

System Programmer Response: Verify that the format and contents of your configuration files are correct. Add the requesting name server to the list of approved name servers if necessary. Restore the configuration files from a backup if the information in the files was inadvertently been lost or corrupted

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6652I approved AXFR from *hostname* **for** ’*zone*’

Explanation: The domain name server received a valid zone transfer request from a secondary name server.

System Action: The domain name server begins sending the resource information in the indicated zone to the secondary name server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6653W Unauthorized request for *domainname* **from** *hostname*

Explanation: The domain name server received a query request from an unapproved TCP/IP host. The requesting system is not in the list of hosts allowed to query information from the zone containing the indicated resource.

The secure network information for a zone is obtained from one or more TXT resource records within the zone with the resource name ″secure_zone″. For example, the secure network information for the zone ″ibm.com″ is contained in the TXT records within the zone file with the name ″secure_zone.ibm.com″.

System Action: The domain name server rejects the received request and continues normal processing.

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

System Programmer Response: Verify that the format and contents of your configuration files are correct. Add the requesting host to the list of approved hosts if necessary. Restore the configuration files from a backup if the information in the files was inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information, the proper format of zone files, and how to specify secure zone information.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6654E ns_req: Out Of Memory

Explanation: The domain name server could not forward a query request to other name servers for resolution. The name server is trying to forward the request to other name servers because it did not have the information needed to answer the query. It could not forward the request because it did not have the storage needed to build and send the request.

System Action: The domain name server sends back a response indicating an error occurred while processing the query. The name server continues normal processing.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect other operations.

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

System Programmer Response: No immediate action is required unless the error persists. The next time you start the domain name server, increase the amount of storage available for it.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6655E res_mkquery(*domainname*) failed

Explanation: The domain name server encountered an error while trying to forward a query to another name server. The local name server received the original query request, did not have the answer, and is attempting to forwarded the request to another name server (recursive query processing). Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server continues normal processing. The name server sends back a reply indicating it encountered an error processing the query request.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6656I ns_req: no address for root server

Explanation: The domain name server encountered an error while trying to forward a query to another name server. The local name server received the original query request, did not have the answer, and is attempting to forward the request to another name server (recursive query processing).

The name server does not have any root name servers for the resource class. The query cannot be forwarded to other name servers in the network.

System Action: The domain name server continues normal processing. The name server sends back a reply indicating it encountered an error processing the query request.

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

System Programmer Response: Verify that the format and contents of the configuration files are correct. Add the needed root server definitions to your cache zone file if necessary. Restore the configuration files from a backup if the information in the files was inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information, the proper format of zone files, and how to specify secure zone information.

If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_REQ

Procedure Name: req_query

EZZ6657E named: send of zone information failed: *error information*

Explanation: The domain name server encountered an error while performing a zone transfer. The name server is the primary name server for the zone and is sending a copy of the zone to a secondary or caching name server. The transmission of the contents of the zone to the remote name server failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The remote name server is probably unable to respond to any queries for resources in the zone until the zone transfer is complete.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated. The error was likely caused by a network failure.

See *IBM Open Class Library Reference, Vol. 1* for more information about the file (socket) write functions (for example, fwrite()) and their possible error values.

Source Data Set: NS_REQ

Procedure Name: fwritemsg

EZZ6658W secondary zone '*origin*' expired

Explanation: The domain name server encountered an unexpected problem while processing a query request. The zone containing the requested resource information expired and not been refreshed. The name server is the secondary name server for the zone and needs to get a refreshed copy of the zone from the primary name server. This is usually an automatic procedure. Possible causes are the refresh of the zone failed or the refresh timer value is greater than the expiration timer value.

System Action: The domain name server continues normal processing, but does not use the information in the expired zone. Processing of the query request continues, but the information in the expired zone is not used to formulate an answer. Depending on the configuration, the query may still be resolved correctly (for example, when forwarding it up to the primary name server for resolution), but it may take longer than usual.

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

System Programmer Response: Verify that the format and contents of your configuration files are correct. Specifically, make sure the expiration periods for your secondary zones are equal to or greater than the refresh periods.

If the configuration of your local name server appears to be correct, the probable cause is a previous zone transfer failed. Check the previously logged messages from the domain name server, and use the information in those messages to resolve any existing problems.

To load any configuration changes or retry the zone transfer, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Source Data Set: NS_REQ

Procedure Name: stale

EZZ6659E secondary zone '*origin*' time warp

Explanation: The domain name server detected what appears to be an internal software error while processing a query request. The refresh time stamp for the zone containing the requested resource information is later than the current system time.

System Action: The domain name server continues normal processing, but does not use the information in the zone. The zone is marked as not authoritative since its contents are suspect.

Processing of the query request continues, but the information in the expired zone is not used to formulate an answer. Depending on the configuration, the query may still be resolved correctly (for example, forwarding it up to the primary name server for resolution), but it may take longer than usual.

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

System Programmer Response: To reload the zone information and obtain the correct time stamp values, reload the domain name server zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

If the problem persists, contact the IBM Software Support Center. Supply the information contained in the message and any other related domain name server messages. If possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration*

Reference for information about how to use its debugging options.

Source Data Set: NS_REQ

Procedure Name: stale

EZZ6660I '*rname class type*' **points to a CNAME (***name***)**

Explanation: The domain name server encountered an unexpected problem while building the response to a query request. The indicated resource definition refers to (points) a CNAME resource definition. CNAME definitions are aliases for real resources and should not be directly referred to by other records.

System Action: The domain name server continues normal processing, skipping the invalid CNAME referral.

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

System Programmer Response: Verify that the format and contents of your configuration files are correct. Specifically, verify that other records (for example, A records) do not directly refer to CNAME records.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Source Data Set: NS_REQ

Procedure Name: doaddinfo

EZZ6661E **named: incorrect hostname** *name* **in WLM data from group** *group* **server** *server* **id** *host*

Explanation: The domain name server could not determine the IP address associated with the server application belonging to the indicated group and MVS host. The Work Load Management (WLM) facility does not contain the IP address information, and the name server does not contain any address information for the hostname associated with the server.

System Action: The domain name server continues normal processing, but the IP address of the server cannot be determined from the name server. Any remote systems that wish to use the server must obtain the address by some other means.

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

System Programmer Response: Either the zone data for the sysplex is incomplete and the requested hostname is not defined, or the server specified the hostname incorrectly when it registered. After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Configuration Guide* for more information about how to define server applications to the name server and how to register these applications with WLM.

Source Data Set: WLM_LOAD

Procedure Name: wlm_load

EZZ6663E **start zone transfer (***zone name* **to** *address***) failed, fork:** *error information*

Explanation: The domain name server encountered an error while attempting to initiate a zone transfer. The name server is a primary name server for the zone and is trying to send a new copy of the zone to a secondary or caching-only name server. The name server process that sends the zone information could not be started. The fork() function failed.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The remote name server's resource information for the zone is not refreshed. Queries received by that name server for resources in the zone may fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error and respond as indicated.

After the problem is resolved, you can load the information into the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information.

See *IBM Open Class Library Reference, Vol. 1* for more information about the fork() function and its possible error values.

Source Data Set: NS_REQ

Procedure Name: startxfr

EZZ6664I zone transfer of '*zone name*' **to** *address* **(pid** *process ID***)**

Explanation: The domain name server is initiating a zone transfer of a zone to the indicated name server. The name server is a primary name server for the zone and is trying to send a new copy of the zone to a secondary or caching-only name server.

System Action: The domain name server begins sending the resource information in the indicated zone to the secondary name server.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_REQ

Procedure Name: startxfr

EZZ6665E named: zone transfer failed, connection failed: *error information*

Explanation: The domain name server encountered an error while performing a zone transfer. The name server is the primary name server for the zone and is sending a copy of the zone to a secondary or caching name server. The transmission of the contents of the zone to the remote name server failed. The local name server could not create a TCP connection with the remote name server.

System Action: The domain name server will retry the zone transfer until the retry count is exhausted. The remote name server's resource information for the zone is not refreshed. Queries received by that name server for resources in the zone may fail until the zone is successfully transferred.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated. The error was likely caused by a network failure.

See *IBM Open Class Library Reference, Vol. 1* for more information about the file (socket) open functions (for example, fdopen()) and their possible error values.

Source Data Set: NS_REQ

Procedure Name: startxfr

EZZ6666I Lame server on '*name*' **(in** '*domain*'**?):** *server address***learned from** *address*

Explanation: The domain name server received a query response with no answer. The local name server received the original query request, did not have the answer, and forwarded the request to another name server (recursive query processing). The name server that sent the response back also did not know the answer, but indicated in the response that the local name server is authoritative for the zone or a higher-level zone. The local name server is not authoritative, or else it would never have forwarded the request to the other name server. The other name server should have known this and not indicated that the local name server was authoritative.

The information in the message indicates the resource being queried, the domain being searched, the address of the name server that sent the response, and the address of the name server that supplied the authoritative name server information to that server.

System Action: The local name server assumes the data in the received response is not trustworthy and retries forwarding the query to the other name servers in the network.

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

System Programmer Response: No action is required, although you may want to verify that the remote name server is configured correctly and is working properly.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6667I Response from unexpected source (*address*)

Explanation: The domain name server received an unexpected query response from a name server It received a response from a name server that it did not forward the query to.

System Action: The local name server ignores the response and continues normal processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6668W MAXQUERIES exceeded, possible data loop in resolving (*name*)

Explanation: The domain name server received a query response with no answer. The local name server received the original query request, did not have the answer, and forwarded the request to another name server (recursive query processing). The name server exceeded the number of times it can forward the query to other name servers in the network. It is possible that the domain network system configuration contains a loop of name servers that keep forwarding queries to one another.

System Action: The domain name server sends back a reply indicating the name server encountered an error processing the query request. The reply indicates the name server failed to locate the requested resource.

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

System Programmer Response: Verify that your domain name system configuration is correct. The hierarchical name space for TCP/IP host names should prevent a recursive configuration, but if a name server contains a definition that specifies a name server is authoritative for the wrong zone, a loop can occur.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6669C resp: storage allocation failed

Explanation: The domain name server could not obtain the storage needed to forward a query to another name server. The local name server received the original query request, did not have the answer, and is attempting to forwarded the request to another name server (recursive query processing).

System Action: The domain name server sends back a reply indicating the name server encountered an error processing the query request.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (malloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the malloc() function and its possible error values.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6670E resp: res_mkquery(*name*) failed

Explanation: The domain name server encountered an error while trying to forward a query to another name server. The local name server received the original query request, did not have the answer, and is attempting to forward the request to another name server (recursive query processing). Information contained in messages previously logged by

the domain name server may help identify the cause of the problem.

System Action: The domain name server sends back a reply indicating the name server encountered an error processing the query request.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6671E ns_resp: sendto(*address*) failed: *error information*

Explanation: The domain name server encountered an error while attempting to forward a query to another name server. The local name server received the original query request, did not have the answer, and is attempting to forwarded the request to another name server (recursive query processing). The TCP/IP sendto() function failed, preventing the transmission of the query.

System Action: The domain name server continues processing the query by forwarding the query to the next name server that can answer the question.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the sendto() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6672W Malformed response from *address* **(***message***)**

Explanation: The domain name server received a query request, did not have the answer, and forwarded the request to another name server (recursive query processing). The reply the remove name server sent back is invalid.

System Action: The domain name server continues processing the query by forwarding the query to the next name server that can answer the question.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the remote name server that sent the invalid response. Supply the information contained in the message and any other related domain name server messages. If possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NS_RESP

Procedure Name: ns_resp

EZZ6673W bogus root NS *data* **received from** *address* **on query for '***name***'**

Explanation: The domain name server received a query request, did not have the answer, and forwarded the request to another name server (recursive query processing). The reply the remove name server sent back identified a root name server that the local name server believes is sending invalid or incorrect data.

System Action: The local name server ignores the information related to the bogus entry and continues processing the reply.

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

System Programmer Response: Verify that the configuration of the name servers in your network are correct. If the configuration is correct, contact the IBM Software Support Center or the vendor that supplied the remote name server

that sent the suspect response. Supply the information contained in the message and any other related domain name server messages. If possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NS_RESP

Procedure Name: doupdate

EZZ6674W bogus non-root NS *data* **received from** *address* **on query for '***name***'**

Explanation: The domain name server received a query request, did not have the answer, and forwarded the request to another name server (recursive query processing). The reply the remove name server sent back indicated another name server is authoritative for the resource, but the local name server believes that name server is sending invalid or incorrect data.

System Action: The local name server ignores the information related to the bogus entry and continues processing the reply.

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

System Programmer Response: Verify that the configuration of the name servers in your network are correct. If the configuration is correct, contact the IBM Software Support Center or the vendor that supplied the remote name server that sent the suspect response. Supply the information contained in the message and any other related domain name server messages. If possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Configuration Reference* for information about how to use its debugging options.

Source Data Set: NS_RESP

Procedure Name: doupdate

EZZ6675E send_msg: sendto(*address***) failed:** *error information*

Explanation: The domain name server encountered an error while attempting to forward a query to another name server. The local name server received the original query request, did not have the answer, and is attempting to forward the request to another name server (recursive query processing). The TCP/IP sendto() function failed, preventing the transmission of the query.

This error can also occur when the domain name server is sending a response to a query.

System Action: The domain name server continues processing the query by forwarding the query to the next name server that can answer the question. Even if the query is lost or the response cannot be sent, DNS resolvers will time out waiting for the reply and resend the query to another name server.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the sendto() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NS_RESP

Procedure Name: send_msg

EZZ6678E sysquery: name lookup error on *name*

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a resource. The name server could not obtain the resource information (IP address) for one of the resources defined in its cache, a resource defined in a zone file, or a resource learned from another name server.

System Action: The domain name server continues normal processing, priming the cache by obtaining the resource information for all other entries defined in the Problems may occur if the missing resource information is needed to resolve other query requests or perform other name server operations.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file

from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6679E sysquery: find name server error (*error*) on *name*

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a name server associated with a resource. The error encountered was either a non-existent domain, or a server failure. The name server being searched for was defined in the local name server's cache, was defined in one of its zone files, or was learned as part of the resource information obtained from another name server. The query response received from the network indicated the remote name servers encountered an error while trying to process the query.

System Action: The domain name server continues normal processing. Problems may occur if the missing name server is needed to resolve other query requests or perform other name server operations.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6680E sysquery: storage allocation failed

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a resource. The name server could not obtain the resource information (IP address) for one of the resources defined in its cache, a resource defined in a zone file, or a resource learned from another name server. The name server could not obtain the storage needed to send a query to another name server.

System Action: The domain name server continues normal processing. Problems may occur if the missing resource information is needed to resolve other query requests or perform other name server operations.

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6681E sysquery: res_mkquery(*name*) failed

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a resource. The name server could not obtain the resource information (IP address) for one of the resources defined in its cache, a resource defined in a zone file, or a resource learned from another name server. Information contained in messages previously logged by the domain name server may help identify the cause of the problem.

System Action: The domain name server continues normal processing. Problems may occur if the missing resource information is needed to resolve other query requests or perform other name server operations.

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

System Programmer Response: Use the information contained in this and any related messages to determine the cause of the error and respond as indicated.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6682E sysquery: name server lookup reports danger (*name*)

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for the name server associated with the indicated resource. The name server being searched for was defined in the local name server's cache, was defined in one of its zone files, or was learned as part of the resource information obtained from another name server.

System Action: The domain name server continues normal processing. Problems may occur if the missing name server is needed to resolve other query requests or perform other name server operations.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6683E sysquery: no addresses found for NS (*name*)

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for the indicated name server. The name server being searched for was defined in the local name server's cache, was defined in one of its zone files, or was learned as part of the resource information obtained from another name server.

System Action: The domain name server continues normal processing. Problems may occur if the missing name server is needed to resolve other query requests or perform other name server operations.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6684E sysquery: sendto(*address*) failed: *error information*

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a resource. The name server could not obtain the resource information (IP address) for one of the resources defined in its cache, a resource defined in a zone file, or a resource learned from another name server.

An error while attempting to send the query to another name server. The TCP/IP sendto() function failed, preventing the transmission of the query.

System Action: The domain name server continues normal processing. Problems may occur if the missing resource information is needed to resolve other query requests or perform other name server operations.

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

System Programmer Response: Use the *error information* in this message to determine the cause of this error. Respond as indicated.

See *IBM Open Class Library Reference, Vol. 1* for more information about the sendto() function and its possible error values. The error was likely caused by a network failure.

Source Data Set: NS_RESP

Procedure Name: sysquery

EZZ6685E check_root: Can't find root server.

Explanation: The domain name server encountered an unexpected problem trying to obtain the list of root name servers for the domain name system. The reply received from the network did not identify any root name servers.

System Action: The domain name server continues normal processing. The name server may not be able to respond successfully to most query requests since it cannot forward any unresolved requests to the root servers for resolution.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: check_root

EZZ6686I check_root: *count* **root servers after query to root server less than minimum**

Explanation: The domain name server encountered an unexpected problem trying to obtain the list of root name servers for the domain name system. The reply received from the network contained fewer root name servers than expected.

System Action: The domain name server continues normal processing. The name server may not be able to respond successfully to most query requests since it appears to missing some root server definitions. It may not be able to forward the unresolved requests to the correct root servers for resolution.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: check_root

EZZ6687I No root name servers for class *class*

Explanation: The domain name server encountered an unexpected problem trying to obtain the resource information for a name server. The name server does not have any root name servers identified for the indicated class of resource. The query cannot be forwarded to other name servers in the network.

System Action: The domain name server continues normal processing. The name server may not be able to respond successfully to most query requests since it cannot forward any unresolved requests to the root servers for resolution. Problems may also occur if the missing name server is needed to resolve other query requests or perform other name server operations.

User or Operator Response: Verify that the format and contents of your zone files are correct. Restore the zone file from a backup if the resource records in the file have inadvertently been lost or corrupted.

After the problem is resolved, you can update the domain name server by requesting the domain name server to reload its zone information. See *z/OS Communications Server: IP Diagnosis* for more information about how to reload the zone information and the proper format of zone files.

Tell the system programmer about the error if necessary.

System Programmer Response: Assist the user if necessary. If the configuration of your local name server appears to be correct, verify that the configurations of the other name servers in your network are correct.

Source Data Set: NS_RESP

Procedure Name: findns

EZZ6688W NODATA and data for '*name***' type** *type* **class** *class*

Explanation: The domain name server received an invalid query response from another name server in the network. The response indicated that no data was found for the requested resource, but the answer section in the response is not empty.

System Action: The domain name server ignores the received response and continues normal processing.

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

System Programmer Response: Contact the IBM Software Support Center or the vendor that supplied the remote name server that sent the invalid response. Supply the information contained in the message and any other related domain name server messages. If possible, start the name server with the debug option, re-create the problem, and provide the resulting debug file. See *z/OS Communications Server: IP Diagnosis*for information about how to use its debugging options.

Source Data Set: NS_RESP

Procedure Name: finddata

EZZ6690I dumping name server statistics

Explanation: The domain name server is dumping its statistics to a file.

System Action: The domain name server creates a file containing all current statistic information.

The domain name server does not answer any resource queries until the statistics are stored. Once the dump is complete, normal processing resumes.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_STATS

Procedure Name: ns_stats

EZZ6691I cannot open statistics file, '*file name*'

Explanation: The domain name server encountered an error while attempting to store its current statistics in a file. The name server could not open the file.

System Action: The domain name server resumes normal processing, but the current values of its statistic counters are not stored.

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

System Programmer Response: Verify the domain name server has write access to the indicated directory and files and there is enough file space available to hold the file.

See *IBM Open Class Library Reference, Vol. 1* for more information about the fopen() function and its possible error values.

Source Data Set: NS_RESP

Procedure Name: ns_stats

EZZ6692I finished dumping name server statistics

Explanation: The domain name server finished saving the current values of its statistic counters.

System Action: The domain name server resumes normal processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_STATS

Procedure Name: ns_stats

EZZ6693C nameserFind: storage allocation failed; *error information*

Explanation: The domain name server could not expand its internal cache. It did not have the storage needed to hold all the resource definitions and information known by the name server.

System Action:

The performance of the domain name server after this error occurs is unpredictable. The shortage of available storage will probably affect operations.

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

System Programmer Response: Increase the amount of storage available for the domain name server and restart the name server. The *error information* in this message provides additional information about the cause of the memory allocation failure (malloc()).

See *IBM Open Class Library Reference, Vol. 1* for more information about the malloc() function and its possible error values.

Source Data Set: NS_STATS

Procedure Name: nameserFind

EZZ6694C nameserIncr([*address*], *which*): bad 'which'

Explanation: An internal error occurred within the domain name server. The name server determined that its internal data areas are corrupted.

System Action: The domain name server continues processing. The behavior may be unpredictable, since it appears some of the internal data areas are corrupted.

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

System Programmer Response: Contact the IBM Software Support Center. Supply the information contained in the message. If possible, dump the domain name server's resource data and statistics, and then restart the name server.

See *z/OS Communications Server: IP Diagnosis* for information about how to request a dump of the name server's resource data and how to control the name server.

Source Data Set: NS_STATS

Procedure Name: nameserIncr

EZZ6695I USAGE *time now boot time data*

Explanation: The domain name server is displaying a subset of its statistics.

System Action: The domain name server continues normal processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_STATS

Procedure Name: ns_logstats

EZZ6697I named starting

Explanation: The domain name server is starting.

System Action: The domain name server begins initializing, reading its configuration information and allocating any required system resources.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6698I name server starting

Explanation: Identifies the version of the domain name server being started.

System Action: None

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6699I named could not determine TCPIPjobname, using default of 'INET'

Explanation: The domain name server could not determine the jobname for the TCP/IP stack. NameD will use the default stack name of 'INET'. A subsequent message will be issued with more error information.

System Action: The domain name server continues.

User or Operator Response: None.

System Programmer Response: For named to connect to a particular stack (in a configuration with multiple concurrent stacks), the TCPIPjobname must be set in the appropriate resolver configuration file or data set. See the *z/OS Communications Server: IP Configuration Guide* for a description of the TCPIP.DATA file search order.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6700E named: could not establish affinity with TCPIPjobname *procname*

Explanation: The domain name server failed to connect to the particular stack, procname. The procname, as determined by the system variable TCPIPjobname, must match ″xxxxx″ where ″xxxxx″ is set in the BPXPRMxx member (used to start OMVS) in the SUBFILESYSTYPE NAME(xxxxx) for ENTRYPOINT(BPXTIINT).

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: Define the procname in the BPXPRMxx member under SUBFILESYSTYPE NAME(xxxxx), or change the TCPIPjobname variable in the resolver configuration file to a defined stack name. See the *z/OS Communications Server: IP Configuration Guide* for a description of the TCPIP.DATA file search order.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6701I named established affinity with '*procname***'**

Explanation: The setibmopt() function completed successfully using the TCP/IP stack, procname.

System Action: The domain name server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6702E named: initialization terminated because TCP/IP Base Feature of OS/390 is not enabled.

Explanation: The domain name server failed to start. Prerequisite componenet of OS/390, TCP/IP is not found.

System Action: The domain name server ends. The name server is not available to respond to queries for host names and addresses.

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

System Programmer Response: None

Source Data Set: NS_MAIN

Procedure Name: main

EZZ6703I named: No registrations returned from WLM

Explanation: The domain name server checked the list of servers registered with WLM but the registration count was zero.

System Action: The domain name server continues.

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

System Programmer Response: Register any desired applications and/or the TCP/IP stack.

Source Data Set: WLM_LOAD

Procedure Name: wlm_load

EZZ6704E named: More that one primary balanced zone specified

Explanation: The domain name server encountered more that one primary cluster zone while processing the boot file.

System Action: The domain name server ends.

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

System Programmer Response: Correct the boot file and reload the domain name server.

Source Data Set: NS_INIT

Procedure Name: boot_read

EZZ6705I *application*: **Unable to run as non-swappable, errno=***errno*, **errnojr=***errnojr*

Explanation: This message is issued to the syslog file when *application* is unable to run in a non-swappable state.

application is the name of the application.

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

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: *application* will run in a swappable state.

User or Operator Response: This message might be expected by the system programmer. Notify the system programmer.

System Programmer Response: If you want *application* to run as non-swappable, do one of the following:

- Do not define the RACF facility, ″BPX.STOR.SWAP″ and start the name server from a userid with IUD=0.
- Define the facility and give the appropriate users at least READ access to the facility.

For more information about configuring RACF for *application*, see the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: NS_MAIN

Procedure Name: main

Chapter 7. EZZ7*xxxx* messages

EZZ7001I Starting.

Explanation: The TFTPD daemon is starting.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7002E New daemon fork failed: *error message*

Explanation: Unable to pre-fork a child daemon.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources and retry and/or correct the system error condition.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7003E Daemon fork to background failed: *error message*

Explanation: Unable to become a daemon.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources and retry and/or correct the system error condition.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7004E Error allocating memory for incoming datagram request: *error message*

Explanation: Unable to allocate storage needed to satisfy a request from a client.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources and retry and/or correct the system error condition.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7005E recvfrom - negative length datagram: *error message*

Explanation: The client sent a datagram of negative length. This is a protocol error.

System Action: TFTPD continues.

User or Operator Response: None.

© Copyright IBM Corp. 1994, 2002

System Programmer Response: Verify client adheres to TFTP protocol. If there is no client-side protocol error, contact IBM service.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7006E recvfrom - no datagram to receive

Explanation: The client sent a datagram of zero length. This is a protocol error.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: Verify client adheres to TFTP protocol.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7007E Could not create a thread: *error message*

Explanation: The server was unable to create a thread to satisfy a client request. The error is ignored.

System Action: TFTPD continues.

User or Operator Response: Retry the request.

System Programmer Response: Verify adequate system resources and/or correct the system error condition.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log.

EZZ7008W Ignoring unknown option -*option*

Explanation: An unsupported command line option was used.

System Action: Command line option ignored.

User or Operator Response: None.

System Programmer Response: Verify command line options used when TFTPD was started.

Source Data Set: TFTPDARG

Procedure Name: Store message in system log. Display message on operator console.

EZZ7009E Could not open cache file: *error message*

Explanation: The file containing the list of files to be cached could not be opened.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify cache file specified on command line exists and has proper permissions and/or correct the system error condition.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7010E Cache file entries exceed limit.

Explanation: The file containing the list of files to be cached was opened but contained too many file names.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify cache file has no more than 30 file names listed.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7011E Bad conversion mode for cache file entry.

Explanation: One or more entries in the file containing the list of files to be cached has an invalid conversion mode.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify cache file has proper conversion modes. 'a' means to preconvert to NETASCII, 'b' means do not preconvert.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7012E Error allocating memory for filename: *error message*

Explanation: TFTPD is unable to allocate memory for caching files.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources and/or correct the system error condition.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7013E Error accessing file: *file name*

Explanation: Unable to access a requested file for caching.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify file has global read permissions and is located in a permitted directory. Verify no directory backout (../) exists in the path and filename.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7014E Could not open file (*file name***) for caching:** *error message*

Explanation: Unable to open a requested file for caching.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify file can be opened, correct any system file problems with file and/or correct the system error condition.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7015E Error reading file (*file name***).**

Explanation: Unable to read a requested file for caching.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify file can be read, correct any system file problems with file.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7016E Error allocating memory for file (*file name*): *error message*

Explanation: Unable to allocate memory to store a requested file for caching.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources to store the file and/or correct the system error condition.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7017E Bad cache file entry.

Explanation: The cache file has an improperly formatted entry.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify cache file.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log.

EZZ7018E No files to be cached found.

Explanation: The cache file was read but did not appear to have any files listed in it.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify cache file.

Source Data Set: TFTPDCCH

Procedure Name: Store message in system log. Display message on operator console.

EZZ7019E Error allocating memory for data buffer: *error message*

Explanation: In attempting to send data to or receive data from the client, TFTPD is unable to allocate memory for a data buffer.

System Action: TFTPD terminates the request by sending an ERROR packet to the client.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources and/or correct the system error condition.

Source Data Set: TFTPDOPS

Procedure Name: Store message in system log.

EZZ7020E Send failed for: *packet type*: *error message*

Explanation: TFTPD was unable to send a DATA, ACK, or OACK to the client.

System Action: TFTPD sends an ERR to the client.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS, TFTPDOPT

Procedure Name: Store message in system log.

EZZ7021E recv failed for *packet type*: *error message*

Explanation: TFTPD was unable to receive a DATA, ACK or OACK from the client.

System Action: TFTPD sends an ERR to the client.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS, TFTPDOPT

Procedure Name: Store message in system log.

EZZ7022E retransmit failed for *packet type*: *error message*

Explanation: TFTPD was unable to complete a retransmit for a DATA, ACK, or OACK packet type.

System Action: TFTPD sends an ERR to the client.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS

Procedure Name: Store message in system log.

EZZ7023E send failed, retry number *number*, **for** *packet type*: *error message*

Explanation: TFTPD was unable to retransmit a DATA, ACK, or OACK to the client.

System Action: TFTPD sends an ERR to the client.

User or Operator Response: None

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS

Procedure Name: Store message in system log.

EZZ7024E Could not create socket: *error message*

Explanation: TFTPD was unable to create a socket.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDSCK

Procedure Name: Store message in system log. Display message on operator console.

EZZ7025E bind failed: *error message*

Explanation: TFTPD was unable to bind a port.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error. Verify TFTPD's well-known port (69) is not already in use.

Source Data Set: TFTPDSCK

Procedure Name: Store message in system log. Display message on operator console.

EZZ7026E connect failed: *error message*

Explanation: TFTPD was unable to set up a socket to send and/or receive from a client.

System Action: The request is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDSCK

Procedure Name: Store message in system log. Display message on operator console.

EZZ7027E nak - send error: *error message*

Explanation: TFTPD was unable to send a negative acknowledgment to the client.

System Action: The error is ignored.

User or Operator Response: Retry the request.

System Programmer Response: None.

Source Data Set: TFTPUTL

Procedure Name: Store message in system log. Display message on operator console.

EZZ7028I *client info*: **Status Write request for** *file name*: *result*

Explanation: Result of the client write request validation.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDVLD

Procedure Name: Store message in system log.

EZZ7029I *client info*: **Status Read request for** *file name*: *result*

Explanation: Results of the client read request validation.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDVLD

Procedure Name: Store message in system log.

EZZ7030I Exiting.

Explanation: TFTPD is exiting.

System Action: TFTPD ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPD (main), TFTPDARG

Procedure Name: Store message in system log. Display message on operator console.

EZZ7031E Exiting Abnormally, Error code: *code*

Explanation: TFTPD is exiting because of an internal error condition.

System Action: TFTPD ends.

User or Operator Response: None.

System Programmer Response: Verify adequate resources, restart TFTPD.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7032E Exiting Abnormally, Signal received: *signal code*

Explanation: TFTPD is exiting because a signal was received.

System Action: TFTPD ends.

User or Operator Response: None.

System Programmer Response: Restart TFTPD.

Source Data Set: TFTPD (main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7033E Could not set code page: *codepage*

Explanation: Codepage is not set.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify codepage is available.

Source Data Set: TFTPDSCP

Procedure Name: Store message in system log. Display message on operator console.

EZZ7034E Incorrect Thread Limit, Max Limit: *limit*

Explanation: An invalid thread limit was specified as a command-line argument.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Restart TFTPD with a valid thread limit.

Source Data Set: TFTPDARG

Procedure Name: Store message in system log. Display message on operator console.

EZZ7035E Could not set Thread Limit to *limit*

Explanation: The system call to set the thread limit for a TFTP daemon failed.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Restart TFTPD with a valid thread limit for your system.

Source Data Set: TFTPD (MVS main)

Procedure Name: Store message in system log. Display message on operator console.

EZZ7036E Error opening file stream: *error message*

Explanation: The system call to open a file stream failed.

System Action: The request is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS

Procedure Name: Store message in system log.

EZZ7037E Error reading file stream: *error message*

Explanation: The system call to read from a file stream failed.

System Action: The request is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS

Procedure Name: Store message in system log.

EZZ7038E *client info*: *direction* **ERROR <error code:** *code*, **msg:** *message*>

Explanation: Trace record for ERROR packet being sent or received. If ERROR received then the request is terminated (client initiated termination).

System Action: TFTPD continues; the request terminated.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDOPS, TFTPDOPT, TFTPUTL

Procedure Name: Store message in system log.

EZZ7039I *client info*: **Status Timeout**

Explanation: A request involving the client failed due to network timeouts.

System Action: The request is terminated.

User or Operator Response: None.

System Programmer Response: Verify the network performance. If necessary the time-out value may be increased by recycling TFTPD and using the -t option upon restart.

Source Data Set: TFTPDOPS, TFTPDOPT

Procedure Name: Store message in system log.

EZZ7040E Error writing file stream: *error message*

Explanation: The system call to write to a file stream failed.

System Action: The request is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TFTPDOPS

Procedure Name: Store message in system log.

EZZ7041E *host name*: **options protocol error**

Explanation: A protocol error occurred between the client and server during options negotiation for a request.

System Action: The request is terminated.

User or Operator Response: None.

System Programmer Response: Verify client implements TFTP protocol correctly. If so, contact IBM service.

Source Data Set: TFTPDOPT

Procedure Name: Store message in system log.

EZZ7042E **Error with archive:** *directory name*

Explanation: A specified archive directory name could not be used.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Verify archive directories specified exist, and are world-readable.

Source Data Set: TFTPDARG

Procedure Name: Store message in system log. Display message on operator console.

EZZ7043E **Number of archives specified exceed limit.**

Explanation: More than 20 archive directories were specified.

System Action: TFTPD is terminated.

User or Operator Response: None.

System Programmer Response: Restart TFTPD, specifying 20 or fewer archive directories.

Source Data Set: TFTPDARG

Procedure Name: Store message in system log. Display message on operator console.

EZZ7044I *client info*: *direction* **RRQ <file=***file name***, mode=***mode name***, recognized options:** *option list***>**

Explanation: Trace record for the RRQ packet.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDVLD

Procedure Name: Store message in system log.

EZZ7045I *client info*: *direction* **WRQ <file=***file name***, mode=***mode name***, recognized options:** *option list***>**

Explanation: Trace record for the WRQ packet.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDVLD

Procedure Name: Store message in system log.

EZZ7046I *client info*: **Status Transaction completed successfully**

Explanation: Trace record for successful completion of a RRQ or WRQ transaction.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDREQ

Procedure Name: Store message in system log.

EZZ7047I *client info*: *direction* **ACK <block=***block***>** *more info*

Explanation: Trace record for ACK packet being sent or received.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDOPS, TFTPDOPT

Procedure Name: Store message in system log.

EZZ7048I *client info*: *direction* **DATA <block=***block***,** *size* **bytes>** *more info*

Explanation: Trace record for DATA packet being sent or received.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPDOPS, TFTPDOPT

Procedure Name: Store message in system log.

EZZ7049I *client info*: *direction* **OACK <blocksize=***block size***>** *more info*

Explanation: Trace record for OACK packet being sent.

System Action: TFTPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFTPUTL

Procedure Name: Store message in system log.

EZZ7051I **Starting**

Explanation: The TIMED application is starting.

System Action: TIMED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMED (main)

Procedure Name: Store message in syslog. Display message to op console.

EZZ7052E Exiting abnormally, daemon fork to background failed: *reason*

Explanation: Unable to become a daemon.

System Action: TIMED is terminated.

User or Operator Response: None.

System Programmer Response: Verify adequate system resources and retry and/or correct the system error condition.

Source Data Set: TIMED (main)

Procedure Name: Store message in syslog. Display message to op console.

EZZ7053E Exiting abnormally, invalid option specified: *option*

Explanation: An unsupported command line option was used.

System Action: TIMED is terminated.

User or Operator Response: Verify command line options used when TIMED was started and retry.

System Programmer Response: Verify command line options used when TIMED was started.

Source Data Set: TIMEDARG

Procedure Name: Store message in syslog. Display message to op console.

EZZ7054E Unable to send to client, sendto() error

Explanation: TIMED was unable to send to the client.

System Action: TIMED continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMEDPTR

Procedure Name: Store message in syslog. Display message to op console.

EZZ7055E Exiting abnormally, recvfrom() error: *reason*

Explanation: TIMED was unable to receive from the client.

System Action: TIMED is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TIMEDPTR

Procedure Name: Store message in syslog. Display message to op console.

EZZ7056E Exiting abnormally, socket() error: *errno*

Explanation: TIMED was unable to create a socket. One possible cause of this error is that the TCP/IP stack must be started before TIMED.

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

System Action: TIMED is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error.

Source Data Set: TIMEDPTR

Procedure Name: Store message in syslog. Display message to op console.

EZZ7057E Exiting abnormally, bind() error: *reason*

Explanation: TIMED was unable to bind a port.

System Action: TIMED is terminated.

User or Operator Response: None.

System Programmer Response: Correct the system error. Verify TIMED's well-known port (37) is not already in use.

Source Data Set: TIMEDPTR

Procedure Name: Store message in syslog. Display message to op console.

EZZ7058I Exiting

Explanation: TIMED is exiting.

System Action: TIMED ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMED (main)

Procedure Name: Store message in syslog. Display message to op console.

EZZ7059E Exiting Abnormally, Signal received: *signal*

Explanation: TIMED is exiting because a signal was received.

System Action: TIMED ends.

User or Operator Response: None.

System Programmer Response: Restart TIMED.

Source Data Set: TIMED (main)

Procedure Name: Store message in syslog. Display message to op console.

EZZ7060E Request received from *clientaddr*

Explanation: Indicates that a time request was received from a client.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMEDPTR

Procedure Name: Store message in syslog.

EZZ7061I Time successfully sent to *clientaddr*

Explanation: Indicates that a time request from a client was satisfied.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMEDPTR

Procedure Name: Store message in syslog.

EZZ7062I Daemon started

Explanation: Indicates that the TIMED application successfully became a daemon.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMED (main)

Procedure Name: Store message in syslog. Display message to op console.

EZZ7262 There is a problem reading the configuration file.

Explanation: The DHCP server had a problem accessing or interpreting the configuration file.

System Action: None.

User or Operator Response: None.

System Programmer Response: A previous message may indicate the specific problem. If not, verify the configuration file exists in the proper directory, has the necessary file access permissions and contains valid configuration options. DHCP defaults to locating the DHCPSD.CFG file in the ETC path - make sure the ETC environment variable is set if using this path. Otherwise specify the full path for the configuration file when starting the DHCP server.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7263 USAGE: *name* **[-?] | [-b] | [[-v | -q] [<ConfigFile>]]**

Explanation: Banner Text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7264 *name* **Version** *version* **- IBM DHCP**

Explanation: Banner Text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7265 This program is the DHCP server. This server is used to provide dynamic configuration updates of hosts on the attached networks. The configurations that this server provides are defined in the server's configuration file.

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7266 Where:

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7267 -? Display help message.

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7268 -b Display the program banner.

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7269 -q Execute in quite mode.

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7270 -v Execute in verbose mode.

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7271 <ConfigFile> DHCP Server configuration file. This file defaults to the

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7272 following rules:

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7273 NAME: *name*

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7274 LOCATION: Searches the current working directory, if not there

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7275 the ETC directory (*name*) is searched.

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7276 RELATED COMMANDS: dadmin

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console.

EZZ7277I DHCP Server Initialized at *time*

Explanation: Informational text.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dhcpsd

Procedure Name: Display message to op console. Write message to user log.

EZZ7278 Options:

Explanation: Informational text displayed when running -v ″verbose″ mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: profile

Procedure Name: Display message to op console.

EZZ7279 Mismatched scope parameters detected. You could experience unexpected results.

Explanation: none

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: profile

Procedure Name: Display message to op console.

EZZ7280 Processing will continue without change.

Explanation: none

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: profile

Procedure Name: Display message to op console.

EZZ7281 Request from: *clientid*

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7282 Type:

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7283 DISCOVER

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7284 REQUEST

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7285 DECLINE

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7286 RELEASE

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7287 INFORM

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7288 BOOTP REQUEST

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7289 UNKNOWN MESSAGE TYPE

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7290 Status:

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7291 **IP Addr:** *addr*

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7292 *msg*

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7293 **Client authentication failed - NO REPLY.**

Explanation: none

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7294 **Offering address to the client - REPLY OFFER.**

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7295 **No addresses available for the client - NO REPLY.**

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7296 Offering reserved address to the client - REPLY OFFER.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7297 Previously bound/no addresses available for client - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7298 Offering expired/released address to client - REPLY OFFER.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7299 Expired/released address not available for the client - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7300 No address is available for client (unknown error) - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7301 aborting OFFER.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7302 Requested hostname does not match offered - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7303 Responding with client information - REPLY ACK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7304 Renewing an unused lease - REPLY ACK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7305 Renewing a lease on another server - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7306 Requested address not in address pool - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7307 Renewing an unused lease REPLY NONE.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7308 Requesting an unavailable address - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7309 Requesting an existing lease - REPLY ACK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7310 NAKing existing lease renew - REPLY NAK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7311 Renewing a released lease - REPLY ACK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7312 Released lease is unavailable to renew - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7313 Renewing an expired lease - RESPONSE.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7314 Cannot renew expired lease - NO RESPONSE.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7315 Requesting a reserved address - REPLY ACK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7316 Requesting an unavailable reserved address - REPLY NAK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7317 Offer not accepted by client - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7318 Client requested address lease from wrong server - REPLY NAK.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7319 Attempting to act on a non-existing lease - NO ACTION.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7320 No server was specified - NO ACTION.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7321 Reserved lease declined by client - ADDRESS MARKED DECLINED.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7322 Active lease declined by client - ADDRESS MARKED DECLINED.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7323 Reserved lease released by client - ADDRESS RELEASED.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7324 Active lease released by client - ADDRESS RELEASED.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7325 Server is not configured to support BOOTP - NO REPLY.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7326 Offering a BOOTP client a permanent lease - REPLY OFFER.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7327 Reply cancelled because requested address is not in our pool.

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7328 Validity of network location undetermined - REPLY NONE

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7329 Client moved to another subnet - REPLY NAK

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7330 Client may have received address from another server - REPLY NONE

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7331 Client statement found but address in use - REPLY NONE

Explanation: Informational text displayed when running ″verbose″ (-v) mode.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: replygen

Procedure Name: Display message to op console.

EZZ7350 (User: *User_name***) Unable to access HTML file** *File_name*

Explanation: The HTML file *File_name* could not be accessed by the IBM Network Station Manager. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again. If the file does not exist, reinstall the IBM Network Station Manager licensed program.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcgi.c

Procedure Name: None.

EZZ7351 (User: *User_name***) Required information** *Variable_name* **not found or not valid.**

Explanation: Information required by the IBM Network Station Manager was not found in the HTML form data or was not valid. This might be caused by one of the following:

- Viewing HTML pages from the IBM Network Station Manager without going through the IBM Network Station Manager main screen.
- Using IBM Network Station Manager CGI programs from non-IBM Network Station Manager HTML pages.
- Editing the IBM Network Station Manager HTML files.
- Not allowing the IBM Network Station Manager to control the flow of operation by using the back up button on the browser to return to previous screens.

System Action: None.

User or Operator Response: Do one of the following and try the request again:

- Only use the published URL to access the IBM Network Station Manager functions.
- Do not use IBM Network Station Manager CGI programs from other HTML pages.
- If the HTML files have been edited, reinstall the IBM Network Station Manager licensed program to restore the HTML files to their correct state.
- Restart the IBM Network Station Manager.
- Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcgi.c qytcmscg.c qytcmssc.c

Procedure Name: None.

EZZ7352 (User: *User_name***) Problem found with HTML file** *File_name*

Explanation: The IBM Network Station Manager cannot process the request due to a problem in HTML file *File_name*. This might be caused by one of the following:

- Using IBM Network Station Manager CGI programs from non-IBM Network Station Manager HTML pages.
- Editing the IBM Network Station Manager HTML files.

System Action: None.

User or Operator Response: Do one of the following and try the request again:

- Do not use IBM Network Station Manager CGI programs from other HTML pages.
- If the HTML files have been edited, reinstall the IBM Network Station Manager licensed program to restore the HTML files to their correct state.
- Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmclc.c qytcmpgc.c qytcmscg.c qytcmtcg.c qytcmtgc.c

Procedure Name: None.

EZZ7353 (User: *User_name***) Unable to process data from HTML page.**

Explanation: The IBM Network Station Manager was unable to process the data from the HTML page. This might be caused by one of the following:

- Using IBM Network Station Manager CGI programs from non-IBM Network Station Manager HTML pages.
- Editing the IBM Network Station Manager HTML files.
- An error reading data from the HTML page.

System Action: None.

User or Operator Response: Do one of the following and try the request again:

- Do not use IBM Network Station Manager CGI programs from other HTML pages.
- If the HTML files have been edited, reinstall the IBM Network Station Manager licensed program to restore the HTML files to their correct state.
- If neither of the above apply, retry your request.
- Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmaic.c qytcmccg.c qytcmclc.c qytcmpcg.c qytcmsgc.c qytcmssc.c qytcmtcc.c

Procedure Name: None.

EZZ7354 (User: *User_name***) Error during authentication for user** *User_profile_name*

Explanation: The user name supplied by the server was null. This can occur if the user was not authenticated before the IBM Network Station Manager program is invoked.

System Action: None.

User or Operator Response: Correct the error and try to sign on to the IBM Network Station Manager program again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcgi.c

Procedure Name: None.

EZZ7355 (User: *User_name***) Unable to access required file** *File_name*

Explanation: The required file *File_name* could not be accessed by the IBM Network Station Manager. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again. If the file does not exist, reinstall the IBM Network Station Manager licensed program.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmpgc.c qytcmpsc.c qytcmtcc.c

Procedure Name: None.

EZZ7356 (User: *User_name***) Unable to access system preferences file** *File_name*

Explanation: The system-wide preferences in file *File_name* could not be accessed by the IBM Network Station Manager. The system-wide preferences for the client application being worked with cannot be viewed or changed. They might not be usable by the client application meaning that the system preferences will use their default values as shipped by IBM. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

If the file was damaged, delete the file and restore a previously saved copy or create it again by selecting new system preferences through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmpgc.c qytcmpsc.c

Procedure Name: None.

EZZ7357 (User: *User_name***) Unable to access user preferences file** *File_name*

Explanation: The user preferences in file *File_name* could not be accessed by the IBM Network Station Manager. The user preferences for the client application and user being worked with cannot be viewed or changed. They might not be usable by the client application meaning that the user preferences will use the system values. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

If the file was damaged, delete the file and restore a previously saved copy or create it again by selecting new user preferences through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmpgc.c qytcmpsc.c

Procedure Name: None.

EZZ7358 (User: *User_name***) Unable to update system preferences in file** *File_name*

Explanation: The system-wide preferences in file *File_name* could not be updated by the IBM Network Station Manager. None of the requested changes to the system preferences have been made. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmpsc.c qytcmtcc.c

Procedure Name: None.

EZZ7359 (User: *User_name*) Unable to update user preferences in file *File_name*

Explanation: The user preferences in file *File_name* could not be updated by the IBM Network Station Manager. None of the requested changes to the user preferences have been made. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmpsc.c qytcmtcc.c

Procedure Name: None.

EZZ7360 (User *User_name*) Unable to retrieve list of users.

Explanation: The IBM Network Station Manager was unable to retrieve the requested list of users. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again. Otherwise, enter the specific user name directly in the entry field.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmclc.c

Procedure Name: None.

EZZ7361 (User: *User_name*) Unable to convert program data from one Coded Character Set ID to another.

Explanation: The IBM Network Station Manager encountered an error while converting program data from one Coded Character Set to another. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: This error indicates a system problem in the area of National Language Support. Verify that the NLS support was properly installed.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmccv.C

Procedure Name: None.

EZZ7362 (User: *User_name*) Unable to access system hardware preferences in file *File_name*

Explanation: The system-wide IBM Network Station hardware preferences in file *File_name* could not be accessed by the IBM Network Station Manager. The system-wide preferences in this file cannot be viewed or changed. This file is also accessed by IBM Network Stations and might not be readable by them. IBM Network Stations might still be able to read their default values as shipped by IBM, their workstation-specific preferences, and their user-specific preferences. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

If the file was damaged, delete the file and restore a previously saved copy or create it again by selecting new system hardware preferences through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtcc.c

Procedure Name: None.

EZZ7363 (User: *User_name***) Unable to access workstation or user specific hardware preferences in file** *File_name*

Explanation: The workstation or user specific IBM Network Station hardware preferences in file *File_name* could not be accessed by the IBM Network Station Manager. The workstation or user specific hardware preferences in this file cannot be viewed or changed. The file also might not be readable by the IBM Network Station when it loads preferences for itself or its user. The IBM Network Station might still be able to read the default preferences as shipped by IBM and the system-wide preferences. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

If the file was damaged, delete the file and restore a previously saved copy or create it again by selecting new workstation or user specific hardware preferences through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtcc.c

Procedure Name: None.

EZZ7364 (User: *User_name***) Unable to update workstation preferences in file** *File_name*

Explanation: The workstation preferences in file *File_name* could not be updated by the IBM Network Station Manager. None of the requested changes to the workstation preferences have been made. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtcc.c

Procedure Name: None.

EZZ7368 (User: *User_name***) Required file** *File_name* **contains a statement that is not valid.**

Explanation: The required file *File_name* contains a statement which could not be parsed by the IBM Network Station Manager. The IBM Network Station Manager cannot continue because of the possibility that preference settings will be lost. This file is installed with the IBM Network Station Manager. The IBM Network Station Manager detected a change since installation.

System Action: None.

User or Operator Response: Reinstall the IBM Network Station Manager licensed program.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtcc.c

Procedure Name: None.

EZZ7369 (User *User_name*) does not have the proper authority to access user *Accessed_User_Name*

Explanation: User *User_name* does not have the proper authority to perform administrative functions through the IBM Network Station Manager.

System Action: None.

User or Operator Response: Do one of the following and try the request again.

Contact a valid System Administrator and have them set up user *User_name* as a Systems Administrator.

Use a different user profile when signing on to the IBM Network Station Manager which has the required authority.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcgi.c

Procedure Name: None.

EZZ7371 (User: *User_name*) Unable to access IBM Network Station DNS configuration file *File_name*

Explanation: The DNS configuration file, *File_name*, could not be accessed. This file contains statements that configure the domain name servers and host tables on IBM Network Stations. IBM Network Stations might not be able to connect to remote hosts because of this error. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error given in the message below. Then delete the file if it exists. Use the "Update host table and DNS configuration" button on the Hardware Settings - System Defaults page of the IBM Network Station Manager to create the file again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtnr.c qytcmtsc.c

Procedure Name: None.

EZZ7373 (User: *User_name*) Unable to access startup file *File_name*

Explanation: The application startup data in file *File_name* could not be accessed by the IBM Network Station Manager. The application startup programs, menus, environment variables, or Internet network information cannot be viewed or changed and might not be usable. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

If the file was damaged, delete the file and restore a previously saved copy or create it again by selecting new startup or Internet network information through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmscg.c qytcmssc.c

Procedure Name: None.

EZZ7374 (User: *User_name*) Unable to access or create system startup file *File_name*

Explanation: The system startup file *File_name* could not be accessed or created by the IBM Network Station Manager. In order to save a user startup file, the system startup file must exist. The requested changes to the user startup programs, menus, environment variables, or Internet network information cannot be saved. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

If the system file was damaged, delete the file and restore a previously saved copy or create it again by selecting new startup or Internet network information through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmssc.c

Procedure Name: None.

EZZ7375 (User: *User_name***) Unable to update startup file** *File_name*

Explanation: The application startup information in file *File_name* could not be updated by the IBM Network Station Manager. The requested changes to the startup programs, menus, environment variables, or Internet network information cannot be saved. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmssc.c

Procedure Name: None.

EZZ7376 (User: *User_name***) Corrupted password value found in system preferences file** *File_name*

Explanation: The IBM Network Station Manager encountered a corrupted password setting in the system-wide preferences file (*File_name*). The system-wide preferences in this file have been corrupted and are not viewable. The file also might not be readable by IBM Network Stations.

System Action: None.

User or Operator Response: Data in this file was lost. Delete the file and restore a previously saved copy or create it again by selecting new system hardware preferences through the IBM Network Station Manager.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtcg.c

Procedure Name: None.

EZZ7377 (User: *User_name***) Unable to access required file** *File_name*

Explanation: The required file *File_name* could not be accessed by the IBM Network Station Manager.

System Action: None.

User or Operator Response: If there is a message below, correct the error and try the request again. If there is no message below, the file does not exist. In that case, reinstall the IBM Network Station Manager licensed program.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmscg.c qytcmssc.c

Procedure Name: None.

EZZ7381 (User: *User_name***) Error occurred while determining the national language version.**

Explanation: The IBM Network Station Manager was in the process of determining the correct national language version to present when the error occurred. This might be caused by one of the following:

- The primary language version of the IBM Network Station Manager licensed program was not installed.
- A secondary language version of the IBM Network Station Manager licensed program was installed without having previously installed the primary language version.
- The primary language version of the IBM Network Station Manager does not match the primary language of the system.

System Action: None.

User or Operator Response: Do one of the following and try the request again:

- Reinstall the primary language version of the IBM Network Station Manager licensed program which matches the primary language of the system.
- Reinstall a secondary language version of the IBM Network Station Manager licensed program after installing the primary language version.
- Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmaic.c

Procedure Name: None.

EZZ7383 (User: *User_name***) User** *User* **does not exist.**

Explanation: The user (*User*) is not an existing user name defined on this system. Only those users that have a valid user name can be configured by the IBM Network Station program.

System Action: None.

User or Operator Response: Do one of the following and try the request again:

- **System Administrator:**
  - Specify an existing user name that is valid.
  - Press the Browse button on the HTML screen to display a list of users that are valid and can be configured for the IBM Network Station program. Select a user from the list.
- **Others:**
  - Contact the System Administrator. It is possible that the user name that was authenticated is only valid for access via the Server.
- Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmccg.c

Procedure Name: None.

EZZ7384 (User: *User_name***) IBM Network Station** *Workstation* **not found.**

Explanation: The IBM Network Station name/terminal *Workstation* could not be resolved.

System Action: None.

User or Operator Response: Specify a valid IBM Network Station name. If the name is valid, verify that the domain name server is active.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmccg.c

Procedure Name: None.

EZZ7438 (User: *User_name***) Unexpected condition(s) detected during update Host Table and DNS Configuration:** *Missing_Info*

Explanation: All of the expected information was not available. Information is extracted from the TCPIP.DATA file. See the *z/OS Communications Server: IP Configuration Guide* for a description of the TCPIP.DATA file search order. Refer to the explanations below for the information that might be missing.

No Domain Name
Host Table not updated with a default domain name because one could not be located. No 'search' or 'domain' statement exists from which the default domain name can be identified. On the 'search' statement, the first domain specified is used as the default domain name and the last 'search' or 'domain' statement detected in the file determines the default domain name.

No Name Servers
Host Table not updated with any name servers because none could be located. No 'nameserver' statement exits which identifies the nameserver to be used. One 'nameserver' statement is needed with a single IP address of the nameserver.

System Hardware settings have been updated with other selections made.

System Action: None.

User or Operator Response: If you expect a default Domain Name and/or Name Servers to be updated in the Host Table, correct the error and try the request again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmtcc.c

Procedure Name: None.

EZZ7439 (User: *User_name***) Unable to access Dump file** *File_name*

Explanation: The Dump file *File_name* could not be accessed by the IBM Network Station Manager. See the message below for the specific cause of the error.

System Action: None.

User or Operator Response: Correct the error and try the request again. If the file does not exist, reinstall the IBM Network Station Manager licensed program.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmfdf.c

Procedure Name: None.

EZZ7440 (User *User_name***) Unable to retrieve list of Dump Files from directory** *Dump_Directory_Name***.**

Explanation: The IBM Network Station Manager was unable to retrieve the requested list of dump files from directory *Dump_Directory_Name*. See the message below (if any) for the specific cause of the error. You might not have read permission to the dump directory or there is no valid dump file.

System Action: None.

User or Operator Response: Correct the error and try the request again.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmclc.c

Procedure Name: None.

EZZ7441 **(User:** *User_name***) Unexpected error occurred. Dump saved in file** *Dump_File_Name*

Explanation: Unexpected error occurred. The trace of the CGI is saved in file *Dump_File_Name*

System Action: None.

User or Operator Response: Do one of the following and try the request again.

Try submitting the request again.

Restart the IBM Network Station Manager.

Contact your Network Station Administrator with the Dump file name.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcut.c

Procedure Name: None.

EZZ7442 **(User:** *User_name***) User** *Initiating_User_name* **initiating conversation is not the current user** *User_name*

Explanation: In processing a series of requests, the IBM Network Station Manager detected a request coming from user *User_name* who is not the user that initiated the conversation with the IBM Network Station Manager. All requests on the current session must originate from user *Initiating_User_name*.

System Action: None.

User or Operator Response: Do one of the following and try the request again.

Try submitting the request again.

Restart the IBM Network Station Manager.

Contact your Network Station Administrator.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcgi.c

Procedure Name: None.

EZZ7443 **(User:** *User_name***) Request received from unexpected IP Address (Unexpected:** *Unexpected_IP_Addr***, Expected:** *Expected_IP_Addr***).**

Explanation: The IP Address *Unexpected_IP_Addr* for this request does not match the previous IP Address *Expected_IP_Addr* used during this session. In processing a series of requests, the IBM Network Station Manager detected a request coming from IP address *Unexpected_IP_Addr*. This is not the IP address that initiated the conversation with the IBM Network Station Manager. All requests on the current session must originate from IP address *Expected_IP_Addr*.

System Action: None.

User or Operator Response: Do one of the following and try the request again.

Try submitting the request again.

Restart the IBM Network Station Manager.

Contact your Network Station Administrator.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmcgi.c

Procedure Name: None.

EZZ7444 (User: *User_name***) IBM Network Station Message Log Initialization Failed with RC:** *Return_Code***., Dump file (** *Dump_File_Name* **) created.**

Explanation: An unexpected error occurred during initialization for the IBM Network Station Manager message logging. If possible, a dump file (*Dump_File_Name*) was created.

System Action: None.

User or Operator Response: Try the request again.

If the problem persists, collect the appropriate information and contact your Service Representative.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmccv.c

Procedure Name: None.

EZZ7445 (User: *User_name***) IBM Network Station Manager System Initialization failed with RC:** *Return_Code***., Dump file (** *Dump_File_Name* **) created.**

Explanation: An unexpected error occurred during system initialization for the IBM Network Station Manager. If possible, a dump file (*Dump_File_Name*) was created.

System Action: None.

User or Operator Response: Try the request again.

If the problem persists, collect the appropriate information and contact your Service Representative.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmain.c qytcmcls.c qytcmcsp.c qytcmjgt.c qytcmjst.c qytcmpgt.c qytcmpst.c qytcmsgt.c qytcmsst.c qytcmtgt.c qytcmtst.c

Procedure Name: None.

EZZ7446 Description for message *Unknown_Message* **could not be found.**

Explanation: The IBM Network Station Manager issued a message for which the message detail could not be found in the message catalog.

This could be caused by applying a PTF which requires a message catalog update and the catalog was not updated.

System Action: None.

User or Operator Response: See your Network Station Administrator.

Ensure that the message catalog was updated with the latest message descriptions. If the error persists, collect the appropriate documentation and contact your Service Representative.

System Programmer Response: None.

Source Data Set: qytcmccv.c

Procedure Name: None.

EZZ7447 Unexpected system error occurred. errno= *Errno***, Message:** *description*

Explanation: A system call returned an unexpected error. The error number received is listed above, along with any text which might be available from the system identifying the error.

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

description describes the meaning of the *errno*.

System Action: None.

User or Operator Response: Contact your Network Station Administrator.

Ensure that program temporary fixes (PTFs) included with the IBM Network Station Manager licensed program have been applied. Restart the IBM Network Station Manager and then try the request again.

If the error persists, collect the available documentation and contact your Service Representative.

System Programmer Response: None.

Source Data Set: qytcmccv.c qytcmcgi.c

Procedure Name: None.

EZZ7448 (User: *User_name***) Unexpected error (** *Signal_Name* **) occurred, Dump file (** *Dump_File_Name* **) created.**

Explanation: An unexpected error occurred while running the IBM Network Station Manager. If possible, a dump file (*Dump_File_Name*) was created.

Refer to your system documentation for a description of the signal error (*Signal_Name*) returned.

System Action: None.

User or Operator Response: Do one of the following and try the request again.

If the problem persists, collect the appropriate information and contact your Service Representative.

Message resulted from Process Name / ID: *Process_Name* / *Process_ID*

System Programmer Response: None.

Source Data Set: qytcmccv.c

Procedure Name: None.

EZZ7450I FFST *text*

Explanation: TCP/IP attempted to establish an interface to First Failure Support Technology (FFST) and was unsuccessful. The failure is related to the installation on the operating system.

text is one of the following:

SUBSYSTEM IS NOT INSTALLED
The FFST Subsystem is not installed on your operating system. TCP/IP can be initialized with out it.

INITIALIZATION MODULE IS NOT FOUND
The FFST Subsystem is installed but the initialization module cannot be found.

INTERFACE MODULE IS NOT FOUND
The FFST Subsystem is installed but the interface module could not be found.

TERMINATION MODULE NOT FOUND
The FFST Subsystem is installed but the termination module cannot be found.

Source Data Set: EZBITINI

System Action: TCP/IP initialization continues, but FFST will not be available to support TCP/IP diagnostics.

User or Operator Response: Save the system log for problem determination. Contact the system programmer to diagnose this error.

User or Operator Response: None.

Message Destination: None.

System Programmer Response: Determine the cause of the error, and if FFST support is desired. If so, you must restart TCP/IP if FFST support is desired for diagnostics.

EZZ7452I FFST INITIALIZATION FAILED, RETURN CODE: *return code*

Explanation: FFST experienced an internal error during initialization with a return code provided. TCP/IP can complete its initialization.

return code is the hexadecimal return code from FFST initialization. See the *FFST/MVS FFST/VM Operations Guide* for imformation about FFST Probe Return Codes.

Source Data Set: EZBITINI

System Action: TCP/IP initializes.

User or Operator Response: Contact your system programmer.

System Programmer Response: Determine the cause of the error, and if FFST support is desired. If so, then verify that FFST was installed correctly. You must restart TCP/IP if FFST support is desired for diagnostics.

EZZ7475I ICMP will Ignore Redirects due to Routing Application being Active.

Explanation: The OMPROUTE routing application is active and requires that TCPIP ignore ICMP Redirect packets. The IGNOREREDIRECT option was set dynamically.

System Action: TCPIP continues

User or Operator Response: None

System Programmer Response: None

Source Data Set: EZBIERTE, EZBIESOC

Procedure Name: EZBIEADR, EZBIEISO

EZZ7500I reapchild: waitpid loop: pid=*child-pid***, status=***status*"

Explanation: A SIGCHLD was received but the termination status of the child process cannot be retrieved.

child-pid is the return value from waitpid.

status is the status returned by waitpid.

System Action: Processing continues.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: conf

Procedure Name: reapchild

EZZ7501E low on space (have *space-available***,** *host* **needs** *space-needed* **in** *directory***)**

Explanation: Sendmail requires more disk space in the queue directory.

space-available is the free disk space.

host is the host name.

space-needed is the required disk space.

directory is the queue directory name.

System Action: Processing continues.

User or Operator Response: Verify file system resources. Free up additional disk space.

System Programmer Response: None.

Source Data Set: conf

Procedure Name: enoughdiskspace

EZZ7503I *queue-id*/*seq-num*▪**: log terminated, too many parts**

Explanation: Log entry was too long.

queue-id is the queue id of the current mail queue being processed.

seq-num is the log entry sequence number.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: conf

Procedure Name: sm_syslog

EZZ7505I **unable to write** *pid-file*

Explanation: Sendmail is unable to write to the pid file.

pid-file is the sendmail pid file name.

System Action: Processing continues.

User or Operator Response: Verify file system resources and directory and file permissions.

System Programmer Response: None.

Source Data Set: daemon

Procedure Name: getrequests

EZZ7506I **problem creating SMTP socket**

Explanation: Sendmail failed to bind to and listen on the SMTP server port.

System Action: Program terminates.

User or Operator Response: Verify that a sendmail server is not already running. Verify system resources.

System Programmer Response: None.

Source Data Set: daemon

Procedure Name: opendaemonsocket

EZZ7507I **My unqualified host name (***host***) unknown; sleeping for retry**

Explanation: Sendmail failed to retrieve the canonical form of the local host name.

host is the host's short name.

System Action: Processing continues.

User or Operator Response: Verify DNS configuration and availability.

System Programmer Response: None.

Source Data Set: daemon

Procedure Name: myhostname

EZZ7508I **unable to qualify my own domain name (***host***) -- using short name**

Explanation: Sendmail failed to retrieve the canonical form of the local hostname.

host is the host's short name.

System Action: Processing continues.

User or Operator Response: Verify DNS configuration and availability.

System Programmer Response: None.

Source Data Set: daemon

Procedure Name: myhostname

EZZ7510I user *uid* **attempted to** *action*

Explanation: The specified user attempted the specified action.

uid is the uid of the user who attempted the action.

action is the unauthorized action attempted (″run daemon″ or ″purge host status″).

System Action: Program terminates.

User or Operator Response: Verify that no security breach is being attempted.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7511I daemon invoked without full pathname; kill -1 won't work″

Explanation: Sendmail was invoked with a relative pathname. SIGHUP will not restart the daemon.

System Action: Processing continues.

User or Operator Response: Restart using full pathname, if appropriate.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7512I could not setıug▪id(*uid***,** *gid***)**

Explanation: Sendmail failed to reset the processes uid and gid to the real values, in order to restart.

uid is the uid that could not be set.

gid is the gid that could not be set.

System Action: Program terminates and automatic restart fails.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: main

Procedure Name: sighup

EZZ7513I could not exec *sendmail-name*

Explanation: Sendmail failed to restart itself.

sendmail-name is the name used to invoke sendmail.

System Action: Program terminates and automatic restart fails.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: main

Procedure Name: sighup

EZZ7514I sendmail starting

Explanation: Sendmail is starting in a daemon mode.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7515I queueup: cannot create *filename*, **uid=***uid*

Explanation: Sendmail failed to create queue file for a message.

filename is the name of the queue file that could not be created.

uid is the effective uid of the process.

System Action: Processing continues.

User or Operator Response: Verify file system resources and queue directory permissions.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: queueup

EZZ7516I queueup: cannot lock *filename* **Arguments: queue file name, error description**

Explanation: Sendmail failed to lock the queue file for a message.

filename is the name of the queue file that could not be locked.

System Action: Processing continues.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: queueup

EZZ7517I FAILED to grow WorkList for *directory* **to** *size*

Explanation: Insufficient memory is available for queue processing.

directory is the queue directory name.

size is the required worklist size.

System Action: Processing continues.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: grow_wlist

EZZ7519I *filename*: **WARNING: dangerous write permissions**

Explanation: Configuration file allows group and/or other write access.

filename is the configuration filename.

System Action: Processing continues.

User or Operator Response: Disallow group and other write access for configuration file.

System Programmer Response: None.

Source Data Set: readcf

Procedure Name: readcf

EZZ7522A sendmail terminating on error, rc = *status*

Explanation: Sendmail daemon terminated on an error condition.

status is the return code of the process.

System Action: Program terminates.

User or Operator Response: Restart sendmail daemon.

System Programmer Response: None.

Source Data Set: sysexits

Procedure Name: sm_exit

EZZ7523A sendmail terminating on signal (*signo***)**

Explanation: Sendmail daemon terminated upon receiving the specified signal. *signo* is the number of the signal caught.

System Action: Program terminates.

User or Operator Response: Restart sendmail daemon.

System Programmer Response: None.

Source Data Set: sysexits

Procedure Name: sm_exit

EZZ7524I sendmail terminating

Explanation: Sendmail daemon terminated normally.

System Action: Program terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sysexits

Procedure Name: sm_exit

EZZ7527I usage: mailstats ı-C cffile▪ ı-f stfile▪ -o

Explanation: mailstats was invoked with improper command line options.

System Action: Program terminates.

User or Operator Response: Correct command syntax and retry.

System Programmer Response: None.

Source Data Set: mailstats

Procedure Name: main

EZZ7528I mailstats: could not open config file *filename*

Explanation: The sendmail configuration file could not be opened.

filename is the name of the configuration file.

System Action: Program terminates.

User or Operator Response: Verify configuration file existence and permissions.

System Programmer Response: None.

Source Data Set: mailstats

Procedure Name: main

EZZ7529I mailstats: StatusFile filename too long: *filename***...**

Explanation: The sendmail stats file name is too long.

filename is the first 30 characters of the status file name.

System Action: Program terminates.

User or Operator Response: Save error information for system programmer.

System Programmer Response: Reduce the length of the status file name specified by the sendmail configuration file.

Source Data Set: mailstats

Procedure Name: main

EZZ7530I mailstats: Too many mailers defined, *limit* **max.**

Explanation: The sendmail configuration file defines too many mailers for mailstats to process.

limit is the maximum number of mailers mailstats can process.

System Action: Program terminates.

User or Operator Response: Save error information for system programmer.

System Programmer Response: Reduce the number of mailers defined by the sendmail configuration file.

Source Data Set: mailstats

Procedure Name: main

EZZ7531I mailstats: no statistics file located

Explanation: The sendmail configuration file does not specify a statistics file.

System Action: Program terminates.

User or Operator Response: Save error information for system programmer.

System Programmer Response: Add a StatusFile (S) option to the sendmail configuration file and create file.

Source Data Set: mailstats

Procedure Name: main

EZZ7532I mailstats: could not open statistics file *filename*

Explanation: The sendmail statistics file could not be opened.

filename is the name of the statistics file.

System Action: Program terminates.

User or Operator Response: Verify statistics file existence and permissions.

System Programmer Response: None.

Source Data Set: mailstats

Procedure Name: main

EZZ7533I mailstats: unknown statistics file format (file size changed).

Explanation: This version of mailstats is incompatible with the version of sendmail that created the statistics file.

System Action: Program terminates.

User or Operator Response: Save error information for system programmer.

System Programmer Response: Verify that compatible versions of mailstats and sendmail are installed.

Source Data Set: mailstats

Procedure Name: main

EZZ7541I opendaemonsocket: server SMTP socket wedged: exiting

Explanation: Sendmail repeatedly failed to bind to and listen on the SMTP server port.

System Action: Program terminates.

User or Operator Response: Verify that a sendmail server is not already running. Verify system resources.

System Programmer Response: None.

Source Data Set: daemon

Procedure Name: opendaemonsocket

EZZ7543I sendall: unlink(*filename*): permanent

Explanation: Attempt to unlink the specified queue file failed.

filename is the queue file name.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: deliver

Procedure Name: dup_queue_file

EZZ7544I sendall: link(*oldfile*, *newfile*): permanent Arguments: existing queue file name, new queue file name

Explanation: Attempt to link new file name to existing queue file failed.

oldfile is the existing queue file name.

newfile is the new queue file name.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: deliver

Procedure Name: dup_queue_file

EZZ7546I Can't open /dev/null

Explanation: Attempt to open /dev/null failed.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: envelope

Procedure Name: openxscript

EZZ7547I Can't create transcript stream *filename*

Explanation: Failed to open a stream for the specified transcript file.

filename is the name of the transcript file.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: envelope

Procedure Name: openxscript

EZZ7549I fill_fd: *when*: **cannot open /dev/null**″

Explanation: Attempt to open /dev/null failed.

when indicates when the error occurred. It is either ″startup″ or null (″″).

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: main

Procedure Name: fill_fd

EZZ7551I queueup: cannot create queue temp file *filename*, **uid=***uid* **Arguments: queue file name, effective uid**

Explanation: Attempt to create a queue file failed.

filename is the name of the file that could not be created.

uid is the effective uid of the process.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: queueup

EZZ7552I queueup: cannot create data temp file *filename*, **uid=***uid*

Explanation: Attempt to create a queue data temporary file failed.

filename is the name of the file that could not be created.

uid is the effective uid of the process.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: queueup

EZZ7553I 552 Error writing control file *filename*

Explanation: Error writing to the specified control file.

filename is the name of the control file.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: queueup

EZZ7554I 452 Error writing control file *filename*

Explanation: Error writing to specified control file.

filename is the name of the control file.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: queue

Procedure Name: queueup

EZZ7556I 554 savemail: cannot save rejected email anywhere

Explanation: Failed to save rejected mail.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: savemail

Procedure Name: savemail

EZZ7558I Out of memory!!

Explanation: Insufficient memory available.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: util

Procedure Name: xalloc

EZZ7560I Frozen configurations unsupported

Explanation: Sendmail was invoked with the obsolete frozen configuration (″-bz″) command line option.

System Action: Program terminates after processing command line options.

User or Operator Response: Restart program without frozen configuration option.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7561I Invalid operation mode *mode-flag*

Explanation: Sendmail was invoked with an invalid mode (″-b″ option) specified.

mode-flag is the single character option flag that followed ″-b″ on the command line.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation without invalid option.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7562I More than one ″from″ person

Explanation: Sendmail was invoked with multiple sender addresses (″-f″ or ″-r″ options) specified. At most one sender address should be specified.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with no more than one sender address specified.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7563I Bad hop count (*hop count***)**

Explanation: Sendmail was invoked with an invalid hop count (″-h″ option) specified.

hop count is the hop count that was specified on the command line.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with valid hop count specified.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7564I Invalid -N argument

Explanation: Sendmail was invoked with an invalid Delivery Status Notification (″-N″ option) specified.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with valid Delivery Status Notification option.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7565I Duplicate -R flag

Explanation: Sendmail was invoked with multiple Delivery Status Notification return parameter (″-R″) options specified.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with at most one one Delivery Status Notification return parameter option specified.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7566I Invalid -R value

Explanation: Sendmail was invoked with an unknown Delivery Status Notification return parameter (″-R″) option specified.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with valid Delivery Status Notification return parameter option specified.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7567I Invalid syntax in -V flag

Explanation: Sendmail was invoked with an invalid envelope ID (″-V″) option specified.

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with valid envelope ID option.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7568I Illegal body type *body-type*

Explanation: Sendmail was invoked with an invalid message body type (″-B″ option) specified.

body-type is the argument specified with the ″-B″ command line option

System Action: Program terminates after processing command line options.

User or Operator Response: Retry operation with valid message body type option.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7569I Permission denied

Explanation: The user is not authorized to start sendmail SMTP daemon or to purge host status.

System Action: Program terminates.

User or Operator Response: Retry operation with root authority.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7570I You do not have permission to process the queue

Explanation: The user is not authorized to process the mail queue.

System Action: Program terminates.

User or Operator Response: Retry operation with root authority.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7571I Recipient names must be specified

Explanation: Sendmail was invoked without any recipient addresses.

System Action: Program terminates.

User or Operator Response: Retry operation with recipient(s) specified.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7572I File descriptors missing on startup: *descriptors*

Explanation: The specified file descriptors were not open at startup. If this occurs, it is probably the result of an incorrect programmatic invocation of sendmail.

descriptors is a list of the names of the missing descriptors, which may include ″stdin″, ″stdout″, and ″stderr″.

System Action: Processing continues.

User or Operator Response: Verify system resources.

System Programmer Response: Verify correct programming invocation of sendmail.

Source Data Set: main

Procedure Name: main

EZZ7573I cannot open *filename*

Explanation: Traffic Log file (specified by ″-X″ option) could not be opened. This may caused by an incorrect file name or incorrect file and directory access permissions.

filename is the Traffic Log file name specified by the ″-X″ command line option.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify file name and permissions.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7574I Warning: .cf version level (*config-version-level*) exceeds sendmail version *sendmail-version-level* functionality (*max-config-version-level*)″,

Explanation: The version number specfied in the exceeds that supported by this version of sendmail.

config-version-level is the configuration version level specified by the configuration file.

sendmail-version-level is the version level of sendmail.

max-config-version-level is the maximum configuration level supported by this version of sendmail.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify that the intended configuration file was specified.

System Programmer Response: Correct configuration file.

Source Data Set: main

Procedure Name: main

EZZ7575I No local mailer defined

Explanation: Configuration file does not include a ″local″ mailer definition.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify that intended configuration file was specified.

System Programmer Response: Correct configuration file.

Source Data Set: main

Procedure Name: main

EZZ7576I No prog mailer defined

Explanation: Configuration file does not include a ″prog″ mailer definition.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify that intended configuration file was specified.

System Programmer Response: Correct configuration file.

Source Data Set: main

Procedure Name: main

EZZ7577I No *file* mailer defined

Explanation: Configuration file does not include a ″*file*″ mailer definition.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify that intended configuration file was specified.

System Programmer Response: Correct configuration file.

Source Data Set: main

Procedure Name: main

EZZ7578I No *include* mailer defined

Explanation: Configuration file does not include a ″*include*″ mailer definition.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify that intended configuration file was specified.

System Programmer Response: Correct configuration file.

Source Data Set: main

Procedure Name: main

EZZ7579I QueueDirectory (Q) option must be set

Explanation: Configuration file does not specify a QueueDirectory (Q) option.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify that intended configuration file was specified.

System Programmer Response: Correct configuration file.

Source Data Set: main

Procedure Name: main

EZZ7580I cannot chdir(*queue-directory*

Explanation: Could not change to specified queue directory. Either the specified directory does not exist, or it lacks proper permissions.

queue-directory is the queue directory that was specified.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify directory existence and permissions.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7581I main: cannot stat *queue-directory*

Explanation: Sendmail failed to retrieve information about the queue directory. Sendmail does not have search access to some directory containing the specified queue directory.

queue-directory is the queue directory that was specified.

System Action: Program terminates.

User or Operator Response: Verify directory permissions.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7582I daemon: cannot fork

Explanation: Sendmail could not fork a daemon process.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: main

Procedure Name: main

EZZ7583I setuserenv: putenv(*env-setting*) failed

Explanation: Sendmail failed to set the specified environment variable.

env-setting is the environment variable and value that could not be set.

System Action: Program terminates after processing command line options.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: main

Procedure Name: setuserenv

EZZ7585I cannot open

Explanation: The specified configuration file does not exist, does not have proper permissions set, or is in a directory without proper permissions.

System Action: Program terminates.

User or Operator Response: Verify file existence and file and directory permissions.

System Programmer Response: None.

Source Data Set: readcf

Procedure Name: readcf

EZZ7586I cannot fstat

Explanation: Sendmail does not have search access to some directory containing the specified queue directory.

System Action: Program terminates.

User or Operator Response: Verify directory permissions.

System Programmer Response: None.

Source Data Set: readcf

Procedure Name: readcf

EZZ7587I not a plain file

Explanation: The specified configuration file is not a plain file

System Action: Program terminates.

User or Operator Response: Verify configuration file specified.

System Programmer Response: None.

Source Data Set: readcf

Procedure Name: readcf

EZZ7588I invalid rewrite line "*rewrite-line*" **(tab expected)**"

Explanation: A rewrite rule without a required tab was encountered.

rewrite-line is the line from the configuration file that caused the error.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7589I Inappropriate use of *symbol* on LHS

Explanation: The specified symbol was used incorrectly in the left-hand side of a rewrite rule.

symbol is the improper symbol found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7590I R line: null LHS

Explanation: A rewrite rule with an empty left-hand side was encountered.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7591I replacement $*repl-number* out of bounds

Explanation: The right-hand side of a rewrite rule contained an out-of-bounds replacement.

repl-number is the out-of-bounds replacement number found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7592I Inappropriate use of *symbol* on RHS

Explanation: The specified symbol was used incorrectly in the right-hand side of a rewrite rule.

symbold is the improper symbol found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7593I R line: null RHS

Explanation: A rewrite rule with an empty right-hand side was encountered.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7594I invalid argument to V line: *"version-spec"*

Explanation: An invalid version line was encountered.

version-spec is the invalid version specification found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7595I invalid V line vendor code: *"vendor-code"*

Explanation: A version line with an invalid vendor code was encountered.

vendor-code is the invalid vendor code found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7596I unknown control line *"line"*

Explanation: An invalid control line was encountered.

line is the invalid line found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: readcf

EZZ7597I I/O read error

Explanation: An error occurred while reading the configuration file.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: readcf

Procedure Name: readcf

EZZ7598I too many *command-char* **lines,** *limit* **max**

Explanation: The configuration file contains too many configuration lines of the specified type.

command-char is the single character command code.

limit is the maximum number of such commands allowed.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: toomany

EZZ7599I fileclass: cannot open *filename*

Explanation: The specified file does not exist, does not have proper permissions set, or is in a directory without proper permissions.

filename is the name of the file which could not be opened.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Verify file existence and file and directory permissions.

System Programmer Response: none

Source Data Set: readcf

Procedure Name: fileclass

EZZ7600I Unable to open bulletin directory '*directory***'**

Explanation: The bulletin directory could not be opened.

directory is the name of the bulletin directory

System Action: Processing continues. (No new bulletins are delivered.)

User or Operator Response: Verify that bulletin directory exists and has proper permissions.

System Programmer Response: None.

Source Data Set: pop_bull

Procedure Name: pop_bull

EZZ7601I Unable to open *filename*

Explanation: The user's .popbull file could not be opened.

filename is the full pathname of the user's .popbull file.

System Action: Processing continues. (No new bulletins are delivered.)

User or Operator Response: Verify file and directory permissions.

System Programmer Response: None.

Source Data Set: pop_bull

Procedure Name: pop_bull

EZZ7602I Unable to open bulletin file *filename*

Explanation: The specified bulletin file could not be opened.

filename is the name of the bulletin file.

System Action: Processing continues. (Bulletins including and after the problem bulletin are not delivered.)

User or Operator Response: Verify file permissions.

System Programmer Response: None.

Source Data Set: pop_bull

Procedure Name: CopyOneBull

EZZ7603I Bulletin *filename* **does not start with a valid "From " separator**

Explanation: Bulletin From separator line is malformed.

filename is the name of the bulletin file.

System Action: Processing continues. (Bulletins including and after the malformed bulletin are not delivered.)

User or Operator Response: Correct From separator line in bulletin file.

System Programmer Response: None.

Source Data Set: pop_bull

Procedure Name: CopyOneBull

EZZ7605I Unable to open temporary maildrop '*filename***'**

Explanation: The specified temporary mail file could not be created.

filename is the full pathname of the file that could not be opened.

System Action: Processing continues, but user is unable to successfully log in.

User or Operator Response: Verify the permissions for the temporary maildrop directory (/usr/mail/popper).

System Programmer Response: None.

Source Data Set: pop_dropcopy

Procedure Name: pop_dropcopy

EZZ7606I Unable to open trace file "*filename***"**

Explanation: The specified trace file could not be opened.

filename is the name of the trace file.

System Action: Program terminates.

User or Operator Response: Verify trace file directory and permissions.

System Programmer Response: None.

Source Data Set: pop_init

Procedure Name: pop_init

EZZ7607I Unable to obtain port and IP address of client

Explanation: Popper failed to retrieve client address information.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: pop_init

Procedure Name: pop_init

EZZ7608I Unable to get canonical name of client

Explanation: Reverse DNS search for client domain name failed.

System Action: Processing continues.

User or Operator Response: Verify DNS configuration.

System Programmer Response: None.

Source Data Set: pop_init

Procedure Name: pop_init

EZZ7611I Unable to open communication stream for input

Explanation: Popper failed to open input stream.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: pop_init

Procedure Name: pop_init

EZZ7612I Unable to open communication stream for output

Explanation: Popper failed to open output stream.

System Action: Program terminates.

User or Operator Response: Verify system resources.

System Programmer Response: None.

Source Data Set: pop_init

Procedure Name: pop_init

EZZ7613I Failed attempted login to *username* **from host** *host*

Explanation: An attempted login from the specified host failed.

username is the name of the local user account.

host is the name of the client host.

System Action: Processing continues.

User or Operator Response: Verify that no security breach is being attempted.

System Programmer Response: None.

Source Data Set: pop_pass

Procedure Name: pop_pass

EZZ7614I Possible probe of account *username* **from host** *host*

Explanation: The specified host may be checking for the existence of user accounts.

username is the name of the local user account that may have been probed.

host is the name of the client host that may have been probing.

System Action: Processing continues.

User or Operator Response: Verify that no security breach is being attempted.

System Programmer Response: None.

Source Data Set: pop_quit

Procedure Name: pop_quit

EZZ7615I **Stats:** *username msg-del bytes-del msg-rem bytes-rem*

Explanation: Log of statistics on user maildrop activity.

username is the name of the user for whom statistics are reported.

msg-del is the number of messages deleted from the maildrop.

bytes-del is the number of bytes deleted from the maildrop.

msg-rem is the number of messages remaining in the maildrop.

bytes-rem is the number of bytes remaining in the maildrop.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: pop_updt

Procedure Name: pop_updt

EZZ7700I **name required for mailer**

Explanation: A mailer definition with no mailer name was encountered.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7701I **mailer** *mailer-name*: **`=' expected**

Explanation: The specified mailer definition is missing an expected ″=″.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7702I **mailer** *mailer-name*: **empty path name**

Explanation: The specified mailer definition contains an empty path name (″P=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7703I mailer *mailer-name*: null end-of-line string

Explanation: The specified mailer definition contains an empty end-of-line string (″E=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7704I mailer *mailer-name*: null argument vector

Explanation: The specified mailer definition contains an empty argument (″A=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7705I mailer *mailer-name*: null working directory

Explanation: The specified mailer definition contains an empty working directory (″D=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7706I mailer *mailer-name*: null charset

Explanation: The specified mailer contains an empty default character set (″C=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7707I mailer *mailer-name*: **null user name**

Explanation: The specified mailer contains an user (″U=″) definition with no user name.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7708I readcf: mailer U= flag: unknown user *username*

Explanation: The specified mailer contains an user (″U=″) definition with an invalid user name.

username is the invalid username found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7709I mailer *mailer-name*: **null group name**

Explanation: The specified mailer contains an user (″U=″) definition with no group name.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7710I readcf: mailer U= flag: unknown group *group-name*

Explanation: The specified mailer contains an user (″U=″) definition with an invalid group name.

group-name is the invalid group name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7711I M*mailer-name*: **A= argument required**

Explanation: The specified mailer definition does not contain required argument (″A=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7712I M*mailer-name***: P= argument required**

Explanation: The specified mailer definition does not contain required path name (″P=″) definition.

mailer-name is the name of the invalid mailer.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7713I too many mailers defined (*limit* **max)**

Explanation: The configuration file contains too many mailer definitions.

limit is the maximum number of mailer definitions allowed.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemailer

EZZ7714I readcf: null option name

Explanation: An option line with no option name was encountered.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7715I readcf: unknown option name *option*

Explanation: An option line for an unknown option was encountered.

option is the unknown option name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7716I readcf: ambiguous option name *option-found* **(matches** *option1* **and** *option2***)**

Explanation: An option line with an ambiguous option name was encountered.

option-found is the ambiguous option name found.

option1 is the first option name matched.

option2 is the second option name matched.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7717I Unknown 8-bit mode *mode-flag*

Explanation: An unknown 8-bit mode specification was encountered.

mode-flag is the unknown single-character 8-bit mode specification found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7718I Unknown delivery mode *mode-flag*

Explanation: An unknown delivery mode specification was encountered.

mode-flag is the unknown single-character delivery mode specification found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7719I readcf: option *option***: unknown group** *group-name*

Explanation: A default group option specifying an unknown group was encountered.

option is the group option flag (″g″).

group-name is the unknown group name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7720I readcf: I option value *option_value* **unrecognized**

Explanation: A sendmail ResolverOptions(I) specifying an unknown value was encountered.

option_value is the unknown option value found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7721I readcf: Op line: *option-value* **unrecognized**

Explanation: A privacy option specifying an unknown value was encountered.

option-value is the unknown option value found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7722I readcf: option u: unknown user *username*

Explanation: A default user option specifying an unknown user was encountered.

username is the unknown username found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7723I Invalid queue sort order *"option-value"*

Explanation: A queue sorting order option specifying an unknown value was encountered.

option-value is the unknown option value found.

Explanation: The configuration file contains an invalid option specification.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7724I Invalid NoRecipientAction: *option-value*

Explanation: A NoRecipientAction option specifying an unknown value was encountered.

option-value is the unknown option value found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7725I readcf: option DoubleBounceAddress: value required

Explanation: A DoubleBounceAddress option specifying no value was encountered.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7726I readcf: option RunAsUser: unknown user *username*

Explanation: A RunAsUser option specifying an unknown user name was encountered.

username is the unknown username found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7727I readcf: option RunAsUser: unknown group *group-name*

Explanation: A RunAsUser option specifying an unknown group name was encountered.

group-name is the unknown group name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: setoption

EZZ7728I readcf: config K line: no map name

Explanation: A map definition with no map name was encountered.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemapentry

EZZ7729I readcf: config K line, map *map-name***: no map class**

Explanation: A map definition with no map class was encountered.

map-name is the name of the invalid map definition.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemapentry

EZZ7730I readcf: map *map-name***: class** *class-name* **not available**

Explanation: A map definition with an unknown map class was encountered.

map-name is the name of the invalid map definition.

class-name is the name of the unknown class found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: makemapentry

EZZ7731I invalid ruleset name: *"ruleset"*

Explanation: The specified rule set name is invalid or not defined.

ruleset is the invalid ruleset name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: strtorwset

EZZ7732I bad ruleset *ruleset-number* **(***limit* **max)**

Explanation: An invalid rule set number was specified.

ruleset-number is the invalid ruleset number found.

limit is the maximum ruleset number allowed.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: strtorwset

EZZ7733I bad ruleset definition *"ruleset"* **(number required after `=')**

Explanation: An invalid rule set name definition was encountered.

ruleset is the name of the invalid ruleset definition.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: strtorwset

EZZ7734I bad ruleset number *ruleset-number* **in** *"ruleset"* **(***limit* **max)**

Explanation: A rule set name definition specifying an invalid rule set number was encountered.

ruleset-number is the invalid ruleset number found.

ruleset is the name of the invalid ruleset definition.

limit is the maximum ruleset number allowed.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: strtorwset

EZZ7739I *ruleset***: too many named rulesets (***limit* **max)**

Explanation: The configuration file defines too many named rule sets.

ruleset is the name of problem ruleset.

limit is the maximum number of named rulesets allowed.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: strtorwset

EZZ7735I *ruleset***: ruleset changed value (old** *ruleset-num1***, new** *ruleset-num2***)**

Explanation: The configuration file contains conflicting rule set name definitions.

ruleset is the name of the invalid ruleset definition.

ruleset-num1 is the first ruleset number associated with the named ruleset.

ruleset-num2 is the second ruleset number associated with the named ruleset.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: strtorwset

EZZ7736I settimeout: invalid queuewarn subtimeout *priority*

Explanation: A queuewarn timeout option specifying an invalid priority was found.

priority is the unknown priority name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: settimeout

EZZ7737I settimeout: invalid queuereturn subtimeout *priority*

Explanation: A queuereturn timeout option specifying an invalid priority was found.

priority is the unknown priority name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: settimeout

EZZ7738I settimeout: invalid timeout *option*

Explanation: An invalid timeout option was encountered.

option is the unknown timeout option name found.

System Action: Program terminates after completing configuration processing.

User or Operator Response: Save error messages for system programmer.

System Programmer Response: Correct configuration file.

Source Data Set: readcf

Procedure Name: settimeout

EZZ7800I OMPROUTE starting

Explanation: The OMPROUTE application is starting.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: OMPROUTE

Procedure Name: Store message in syslog. Write message to user log.

EZZ7801I Deleting stack route to *destination*, **mask** *mask* **via** *gateway*, **link** *link*, **metric** *metric*, **type** *type*

Explanation: OMPROUTE is deleting the specified route from the stack's route table. This is a route that OMPROUTE discovered using the OSPF or RIP protocol or as the result of a directly connected interface. The route is no longer available.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7802I OMPROUTE invalid option specified: *option*

Explanation: An unsupported command line option was used. *option* is the invalid option.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Verify command line options used when OMPROUTE was started and retry.

Source Data Set: OMPROUTE

EZZ7803I OMPROUTE, *function* **errno=***errno***:***description***, errno2=***errnojr*

Explanation: The indicated function failed with the indicated error.

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

description describes the error.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Correct the system error reported.

Source Data Set: many

EZZ7804I OMPROUTE exiting

Explanation: The OMPROUTE application is exiting normally.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Restart OMPROUTE if desired.

Source Data Set: OMPROUTE

Procedure Name: Store message in syslog. Write message to user log.

EZZ7805I OMPROUTE exiting abnormally - RC(*returncode***)**

Explanation: Application is exiting abnormally with the specified return code:

1 Exited due to unrecoverable error

3 Exited due to receipt of terminating signal

4 Exited due to invalid startup parameter passed to OMPROUTE

5 Exited due to failure to add route to TCP/IP stack's route table

6 Exited due to failure to obtain storage using malloc

7 Exited due to the TCP/IP stack going down

8 Exited while initializing the PSA environment

9 Exited while initializing the log/trace facility

10 Exited due to OMPROUTE not being APF authorized

11 Exited while initializing the OSPF and INFORMATIONAL sockets

12 Exited while configuring the OMPROUTE application

13 Exited while obtaining the TCP/IP stack's route table

14 Exited while initializing the RIP protocol

15 Exited while initializing the OSPF protocol

16 Exited while dynamically adding an OSPF interface

17 Exited due to failure to delete route from TCP/IP stack's route table

18 Exited due to failure to change route in TCP/IP stack's route table

19 Exited while initializing the RIP socket

20 Exited due to failure to update TCP/IP stack's BSD Routing Parameters

21 Exited due to error establishing thread attributes

22 Exited while attempting to start subagent thread

23 Exited because of inability to originate a router LSA. For more information about this condition, see the information about dynamic VIPAs and routing protocols in *z/OS Communications Server: IP Configuration Guide*.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Take necessary corrective action based upon specified return code and any preceding messages indicating the cause of the abnormal exit. Restart OMPROUTE if desired.

Source Data Set: OMPROUTE

EZZ7806I Changing stack route to *destination*, **mask** *mask* **via** *gateway*, **link** *link*, **metric** *metric*, **type** *type*

Explanation: OMPROUTE is updating information about a route in the stack's routing table. This is a route that OMPROUTE discovered using the OSPF or RIP protocol or as the result of a directly connected interface. The characteristics of the route (for example, metric) have changed.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7807I Abnormal termination - out of storage

Explanation: OMPROUTE requested storage for a control block or buffer and that request failed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Resolve the storage shortage.

Source Data Set: OMPROUTE

EZZ7808I Could not determine TCPIP jobname, using default of 'INET'

Explanation: The TCPIP JOBNAME parameter was not found in the TCP/IP profile so the default JOBNAME of ″INET″ is used. This value is ignored in a single-stack environment.

System Action: OMPROUTE continues, using the default JOBNAME of ″INET″.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORINI

EZZ7809I *outputstring*

Explanation: Used for displaying responses to the DISPLAY command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7810I Route not deleted from stack routing table - *reason*

Explanation: A dynamic route was not deleted from the stack's routing table for the specified reason.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7811I Could not establish affinity with *TCPIPjobname*, **errno=***errno*:*description*, **errno2=***errnojr*

Explanation: An attempt to establish affinity with the specified TCP/IP stack failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Correct the system error reported. A possible cause is the TCP/IP stack name being incorrectly specified in the resolver configuration file.

Source Data Set: EZAORINI

EZZ7812I Could not obtain stack interface flags, ioctl errno=*errno*:*description*, **errno2=***errnojr*

Explanation: An attempt to obtain the flags for a TCP/IP interface failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORCFG

EZZ7813I Could not obtain stack interface broadcast address, ioctl errno=*errno***:***description***, errno2=***errnojr*

Explanation: An attempt to obtain the broadcast address for a TCP/IP interface failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: None.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORCFG

EZZ7814I Unable to create socket type *type***, errno=***errno***:***description***, errno2=***errnojr*

Explanation: An attempt to create a socket of the specified type failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Attempt to correct the system error reported. A possible cause is the TCP/IP stack not being started. If problem persists, there is a possible internal error, contact the IBM Software Support Center .

Source Data Set: EZAORCFG, EZAORINI, EZAORRTI

EZZ7815I Socket *socket* **bind to port** *port***, address** *address* **failed, errno=***errno***:***description***, errno2=***errnojr*

Explanation: An attempt to bind to the specified port failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORINI

EZZ7816I Unable to set option *option* **for** *type* **socket** *socket***, errno=***errno***:***description***, errno2=***errnojr*

Explanation: An attempt to set the specified option on the specified socket failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORINI, EZAORRTI, H390MCAS

EZZ7817I Using *type* **OSPF protocol** *protocol*

Explanation: The specified protocol number is being used for OSPF communication.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORINI

EZZ7818I Unable to listen on socket *socket*, **errno=***errno*:*description*, **errno2=***errnojr*

Explanation: An attempt to listen on the specified socket failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: None.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORRTI

EZZ7819I Invalid value for *keyword* **coded on** *type* **statement**

Explanation: The value coded for the specified keyword on the specified configuration statement in the OMPROUTE configuration file was invalid.

System Action: If the keyword has a default value, the default is taken. Otherwise, OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Correct the value coded for the specified keyword in the OMPROUTE configuration file.

Source Data Set: EZAORCFG, EZAORYAC

EZZ7820I Required parameter for *keyword* **not coded on** *type* **statement**

Explanation: A required keyword for which no default can be taken was not coded on the specified configuration statement in the OMPROUTE configuration file.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Add the required keyword to the appropriate configuration statement in the OMPROUTE configuration file.

Source Data Set: EZAORYAC

EZZ7821I Ignoring duplicate *type* statement

Explanation: Found a duplicate of the specified configuration statement in the OMPROUTE configuration file.

System Action: The duplicate statement is ignored.

User or Operator Response: None.

System Programmer Response: Remove the duplicate statement from the OMPROUTE configuration file.

Source Data Set: EZAORCFG, EZAORYAC

EZZ7822I Could not find configuration file

Explanation: The OMPROUTE configuration file could not be found

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Check the search hierarchy for the existence of the OMPROUTE configuration file. Code the configuration file if one does not exist.

Source Data Set: EZAORYAC

EZZ7825I OMPROUTE unrecoverable error (*error*)

Explanation: OMPROUTE is exiting due to the specified unrecoverable error.

System Action: OMPROUTE exits.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: LOGTRACE

EZZ7826I Unable to accept connections on socket *socket*, errno=*errno*:*description*, errno2=*errnojr*

Explanation: An attempt to accept a connection on the specified socket failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORRTI

EZZ7827I Adding stack route to *destination*, mask *mask* via *gateway*, link *link*, metric *metric*, type *type*

Explanation: OMPROUTE is adding the specified route to the stack's route table. This route may have been discovered using OSPF or RIP protocols or may be the result of a directly connected interface.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7828I Error adding/deleting/changing stack rte, return code *retcode*, **ioctl errno=***errno*:*description*, **errno2=***errnojr*

Explanation: An attempt to add, delete, or change a route in the stack's route table failed for the specified reason.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: Depending on the type and severity of the errors, OMPROUTE may terminate or continue processing. For system errors, OMPROUTE will terminate. If OMPROUTE determines the error is only for a particular route, it will continue processing.

User or Operator Response: Determine if the error is caused by a bad router or other network error. Correct the failing device.

System Programmer Response: If OMPROUTE terminates, review the error code description to determine the cause of the problem.

Source Data Set: EZAORMUP, EZAORRTI

EZZ7829I Route not changed in stack routing table - *reason*

Explanation: A dynamic route was not changed in the stack's routing table for the specified reason.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7830I *error* **at line** *line* **of OMPROUTE configuration file processing** *token*

Explanation: The specified error occurred on the specified line in the OMPROUTE configuration file. The error was encountered on or just previous to the specified token.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Correct the error in the OMPROUTE configuration file.

Source Data Set: EZAORLEX

EZZ7831I Global configuration

Explanation: This message precedes the display of the OMPROUTE OSPF configuration in response to the F proc,OSPF,LIST,ALL command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7832I Area configuration

Explanation: This message precedes the display of the OMPROUTE OSPF area configuration in response to the F proc,OSPF,LIST,AREAS command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7833I Interface configuration

Explanation: This message precedes the display of the OMPROUTE OSPF interfaces configuration in response to the F proc,OSPF,LIST,INTERFACES command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7834I Neighbor configuration

Explanation: This message precedes the display of the OMPROUTE OSPF neighbor configuration in response to the F proc,OSPF,LIST,NEIGHBORS command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7835I NBMA configuration

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7836I Virtual link configuration

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCFG

EZZ7837I Could not obtain stack interface index, ioctl errno=*errno***:***description***, errno2=***errnojr*

Explanation: The attempt to obtain the list of interfaces defined to the TCP/IP stack failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORCFG

EZZ7838I Using configuration file: *filename*

Explanation: The specified configuration file is being used to configure OMPROUTE.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORYAC

EZZ7839I *type* **receive thread terminates**

Explanation: The specified OMPROUTE thread is terminating. Communications over this thread are terminated.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Actions determined by preceding messages indicating the cause of the thread termination.

Source Data Set: EZAORIRT, EZAORMFY, EZAORORT, EZAORRRT

EZZ7840I sendto() error, errno=*errno***:***description***, errno2=***errnojr*

Explanation: The indicated error occurred while attempting to send a packet of data to an adjacent router.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: If error is due to TCP/IP stack going down, OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: This may be a temporary condition caused by the fact that informational socket packets have not yet been processed. If this message appears repeatedly, contact the IBM Software Support Center .

Source Data Set: INRIPOUT

EZZ7841I *type* **receive socket closed unexpectedly**

Explanation: A TCPIP socket used for communications by OMPROUTE closed or timed out.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Investigate possible problems with TCPIP. Restart OMPROUTE.

Source Data Set: EZAORIRT, EZAORORT, EZAORRRT

EZZ7842I *type function* **error, errno=***errno***:***description***, errno2=***errnojr*

Explanation: The indicated function failed with the indicated error. *errno* is the system error code.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: If the error is due to the TCP/IP stack going down, OMPROUTE ends. Otherwise, processing continues.

User or Operator Response: None.

System Programmer Response: Correct the system error reported.

Source Data Set: EZAORIRT, EZAORORT, EZAORRRT

EZZ7843I RIP Configuration

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRPCNFG

EZZ7844I RIP Route Acceptance

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7845I Established affinity with *TCPIPjobname*

Explanation: OMPROUTE established affinity with the specified TCP/IP stack.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORINI

EZZ7846I Established affinity with single stack (Common INET is not defined)

Explanation: OMPROUTE always binds itself to a specific stack. This is determined by the TCPIPJOBNAME parameter in the TCPIP.DATA data set. The TCPIP.DATA data set is found according to the search order described in the *z/OS Communications Server: IP Configuration Reference*.

This may be set by using the OMPROUTE start procedure or directly using the export command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORINI

EZZ7847I Routing Table

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the *z/OS Communications Server: IP Configuration Reference* section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INCON

EZZ7848I Area Summary

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCON

EZZ7849I Interfaces

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCON

EZZ7850I Interface Details

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCON

EZZ7851I Neighbor Summary

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7852I Neighbor Details

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7853I Area Link State Database

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7854I Link State Database Size

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7855I OSPF Routers

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7856I OSPF Statistics

Explanation: This message is produced in response to a DISPLAY TCPIP,,OMPROUTE command. See the section about the DISPLAY DISPLAY TCPIP,,OMPROUTE command in the *z/OS Communications Server: IP Configuration Reference* for more information.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

EZZ7857I *error* **(***text***) on OMPROUTE console command**

Explanation: The specified error occurred while processing an OMPROUTE DISPLAY or MODIFY command.

System Action: The OMPROUTE DISPLAY or MODIFY command is ignored.

User or Operator Response: None.

System Programmer Response: Verify the syntax of the DISPLAY or MODIFY command and reissue.

Source Data Set: EZAORLEX

EZZ7858I Unable to send debug and/or trace output to debug destination

Explanation: OMPROUTE was unable to access the debug destination for storage of debug and/or trace output.

System Action: OMPROUTE continues, unable to write debug and trace information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORYAC, OMPROUTE

EZZ7859I RIP Interfaces

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRPCON

EZZ7860I RIP Interface Details

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRPCON

EZZ7861I Sink network discarding packet to *destination*

Explanation: An OSPF or RIP packet, intended for the specified destination, is being discarded due to the lack of an active interface to the destination.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NBMA

EZZ7862I Received *type* **interface** *name*

Explanation: OMPROUTE learned of a status change of the specified type for the specified TCP/IP interface.

System Action: OMPROUTE makes any necessary changes to its processing based upon the status change. Also, any necessary changes are made to routes that use this interface.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMII

EZZ7863I Received *type* **route to** *destination*

Explanation: The status of the specified route, defined by a GATEWAY statement in the TCPIP profile, changed.

System Action: OMPROUTE makes any necessary changes to its internal route table based upon the status change.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMII

EZZ7864I Deleting all stack routes to *destination*, mask *mask*

Explanation: OMPROUTE is deleting all routes to the specified destination/mask from the TCP/IP stack's route table. This is occurring either because the route no longer exists or because it changed (in which case the new route is added following the deletion).

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7865I Class mask *class_mask* being used for interface *interface*

Explanation: The specified interface was not configured in the OMPROUTE configuration file using an OSPF_Interface, RIP_Interface, or Interface configuration statement. Therefore, a default mask is being used based upon the network class.

System Action: The default (class) mask is used for the interface. The network that can be accessed with this interface is computed by ANDing the interface address with this mask. A route to the resulting network (by using this interface) is added to the TCP/IP stack's route table.

User or Operator Response: None.

System Programmer Response: If the specified mask is not desired for the specified interface, modify the OMPROUTE configuration file to add an OSPF_Interface, RIP_Interface, or Interface configuration statement for the interface.

Source Data Set: INCONF

EZZ7866I OMPROUTE MODIFY command accepted

Explanation: A MODIFY command was received by OMPROUTE.

System Action: The MODIFY command is processed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORYAC, SPFCON

EZZ7867I Invalid *type* value specified on OMPROUTE console command

Explanation: An invalid value of the specified type was entered on a DISPLAY or MODIFY command received by OMPROUTE

System Action: The DISPLAY or MODIFY command is ignored.

User or Operator Response: None.

System Programmer Response: Verify the syntax of the DISPLAY or MODIFY command and reissue.

Source Data Set: EZAORYAC, INCON, INRPCON, SPFCON

EZZ7868I Dead Router and DB_Exchange Intervals must be greater than Hello Interval on *type* statement, using defaults

Explanation: A configuration statement of the specified type was encountered in the OMPROUTE configuration file on which either the Dead Router Interval or the DB Exchange Interval is less than or equal to the Hello Interval. This is an invalid configuration.

System Action: OMPROUTE continues, using the default values for Dead Router Interva, DB Exchange Interval,l and Hello Interval for the interface. The OMPROUTE display command (DISPLAY TCPIP,<tcpipjobname>,OMPROUTE,OSPF,LIST, type, where type=INTERFACES or VLINKS) can be used to determine the default values.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file to increase either the Dead Router Interval or the DB Exchange Interval, or decrease the Hello Interval. See the *z/OS Communications Server: IP Configuration Reference* descriptions of the OSPF configuration statements.

Source Data Set: EZAORYAC

EZZ7869I OMPROUTE configuration file must contain OSPF or RIP interface statements

Explanation: OMPROUTE found no OSPF_Interface or RIP_Interface configuration statements in the OMPROUTE configuration file.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file, adding the appropriate OSPF_Interface and/or RIP_Interface configuration statements.

Source Data Set: EZAORYAC

EZZ7870I OSPF and RIP interface statements for *ipad* (*name*) have different values for *kwrd*

Explanation: OMPROUTE found conflicting keyword values on the configuration interface statements (OSPF_Interface, RIP_Interface, or Interface). The specified IP address and interface name have a different value on the specified keyword. This is an invalid configuration. For example, an OSPF_INTERFACE and RIP_INTERFACE statement for the same actual interface specify different MTU sizes.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file to correct the problem.

Source Data Set: EZAORCFG

EZZ7871I No matching interface statements for *ipad* (*name*)

Explanation: No matching OSPF_Interface, RIP_Interface, or Interface statement was found for the specified TCP/IP interface.

System Action: OMPROUTE continues. The specified interface will not be used by the OSPF and RIP protocols. Also, the class mask will be used in calculating the network route to be added to the route table.

User or Operator Response: None.

System Programmer Response: If it is desired that the specified interface be used by the OSPF and/or RIP protocol, modify the OMPROUTE configuration file to add an OSPF_Interface and/or RIP_Interface configuration statement. Otherwise, add an Interface configuration statement if the class mask should not be used in calculating the network route to be added to the route table.

Source Data Set: EZAORCFG

EZZ7872I OMPROUTE found another routing application already active

Explanation: OMPROUTE was unable to allocate required TCP/IP resources. These resources are only used by routing applications. Another instance of OMPROUTE or another routing application is running on the TCP/IP stack.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Stop the active routing application and restart OMPROUTE.

Source Data Set: EZAORINI

EZZ7873I Required parameter *parm* **missing on OMPROUTE DISPLAY command**

Explanation: The specified required parameter was not provided on a DISPLAY command received by OMPROUTE.

System Action: The DISPLAY command is ignored.

User or Operator Response: None.

System Programmer Response: Verify the syntax of the DISPLAY command and reissue.

Source Data Set: EZAORYAC

EZZ7874I Route Expansion

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INCON

EZZ7875I No default route defined

Explanation: Either there is no default route configured in the OMPROUTE configuration file or the default route configured could not be started. This is probably caused by the next hop interface for that route being inactive.

System Action: OMPROUTE continues with no default route configured.

User or Operator Response: None.

System Programmer Response: Activate the next hop interface and the default route will become active.

Source Data Set: EZAORRTI

EZZ7876I -- *protocol* **Packet Sent ------ Type:** *type version*

Explanation: A packet, of the specified type and version, was sent by the specified protocol.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7877I -- *protocol* **Packet Received -- Type:** *type version*

Explanation: A packet, of the specified type and version, was received by the specified protocol.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7878I *formatstring*

Explanation: Used for displaying formatted packet output.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7879I *reason* **multicast group** *group* **on interface** *interface*

Explanation: OMPROUTE is joining or leaving the specified multicast group on the specified interface. These groups are used for RIPv2 and OSPF protocols.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: H390MCAS

EZZ7880I LSA Details

Explanation: None.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFCON

EZZ7881I *field* **required for point to point link** *name*

Explanation: OMPROUTE found an OSPF_Interface, RIP_Interface or Interface configuration statement for the specified interface, which is a point to point link. The specified keyword is required for a point to point link and it was not provided.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file, adding the required keyword.

Source Data Set: EZAORCFG

EZZ7882I Processing static route from stack, destination *dest*, **mask** *mask*, **gateway** *gw*

Explanation: The specified route is defined using a TCPIP GATEWAY statement. OMPROUTE learned of this route during initialization. OMPROUTE cannot override or modify this route. Consider removing this GATEWAY statement while using OMPROUTE, allowing OMPROUTE to dynamically learn and manage the route.

System Action: OMPROUTE will add the route to its internal route table and will never modify the route or add a route to the same destination.

User or Operator Response: None.

System Programmer Response: If possible, remove this GATEWAY statement while using OMPROUTE, allowing OMPROUTE to dynamically learn and manage the route.

Source Data Set: EZAORRTI

EZZ7883I Processing interface from stack, address *addr*, **name** *name*, **index** *index*, **flags** *flags*

Explanation: The specified interface is defined to the TCP/IP stack. OMPROUTE learned of this interface during initialization.

flags are the interface capability flags and are a hexidecimal sum of the following values:

0X'01' The interface is up.

0X'02' The interface is broadcast capable.

| **0X'04'** The interface driver is in debug mode.

| **0X'08'** The interface is in loopback only mode.

| **0X'10'** The interface is a Point-to-Point interface.

| **0X'20'** The interface does not support Trailer encapsulation.

| **0X'40'** The interface is running.

| **0X'80'** The interface is ARP incapable.

| **0X'100'** The interface is in promiscuous mode.

| **0X'200'** The interface is receiving all multicast packets.

| **0X'400'** The interface is multicast capable.

| **0X'800'** The interface is point-to-multipoint.

| **0X'1000'** The interface supports Token Ring bridging.

| **0X'2000'** The interface supports extended SAP.

| **0X'4000'** The interface is a VIPA.

| For example, an active VIPA interface often appears as 4041 (0x4000 + 0x40 + 0x01).

Note: These interface flag values are set by the TCP/IP stack and passed to OMPROUTE.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORYAC

| **EZZ7884I RouterID *id* not a configured OSPF interface, taking default**

| **Explanation:** OMPROUTE found the RouterID configuration statement in the OMPROUTE configuration file. The
| specified ID is not a configured OSPF interface.

| **System Action:** OMPROUTE ignores the RouterID configuration statement and selects one of the configured OSPF
| interfaces as the Router ID.

| **User or Operator Response:** None.

| **System Programmer Response:** Modify the OMPROUTE configuration file, providing a configured OSPF interface
| as the Router ID.

| **Source Data Set:** EZAORYAC

EZZ7885I Route not added to stack routing table - *reason*

Explanation: A dynamic route was not added to the stack's routing table for the specified reason.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ7886I Not connected to area specified on OMPROUTE DISPLAY command

Explanation: An OMPROUTE DISPLAY command was received which contains the specified area ID. This area ID is not a configured area ID.

System Action: The DISPLAY command is ignored.

User or Operator Response: None.

System Programmer Response: Reenter the DISPLAY command with a corrected area id.

Source Data Set: SPFCON

EZZ7887I OMPROUTE is not APF authorized

Explanation: An attempt was made to start the OMPROUTE application, but the application is not APF authorized. APF authorization is required to execute OMPROUTE.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: APF authorize the OMPROUTE application.

Source Data Set: OMPROUTE

EZZ7888I *one* **statement ignored, conflicts with previous** *two* **statement**

Explanation: OMPROUTE found a configuration statement in the OMPROUTE configuration file that conflicts with a previous configuration statement.

System Action: OMPROUTE continues, ignoring the specified configuration statement.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file to resolve the conflict.

Source Data Set: EZAORYAC

EZZ7889I *msgstring*

Explanation: Used for displaying the results of CTRACE initialization. A zero return code and reason code indicates that initialization completed successfully.

System Action: None.

User or Operator Response: None.

System Programmer Response: If CTRACE did not complete successfully, resolve cause of error.

Source Data Set: OMPROUTE

EZZ7890I *kwd* **ignored when** *stmt* **statement is a wildcard**

Explanation: OMPROUTE found the specified wildcard statement in the OMPROUTE configuration file with the specified keyword provided. This keyword is meaningless when the statement is a wildcard.

System Action: OMPROUTE continues, ignoring the keyword provided.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file, removing the superfluous keyword.

Source Data Set: EZAORYAC

EZZ7891I Multiple *stmt* **statements for subnet** *sub* **have** *kwd* **coded as primary**

Explanation: OMPROUTE found more than one of the specified configuration statement in the OMPROUTE configuration file for the specified subnet. Each of these statements has the specified keyword coded as ″primary″.

System Action: OMPROUTE continues, ignoring the ″primary″ setting on all but one of the configuration statements.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file, ensuring that only one of the statements for the specified subnet have the specified keyword coded as ″primary″.

Source Data Set: EZAORCFG

EZZ7892I Error updating stack BSD Routing Parms, return code *retcode*, **ioctl errno=***errno*:*description*, **errno2=***errnojr*

Explanation: An attempt to update the stack's BSD Routing Parameters failed for the specified reason.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORMUP

EZZ7893I IP address *.*.*.* not valid on Interface statement, statement ignored

Explanation: OMPROUTE found an Interface configuration statement in the OMPROUTE configuration file with an IP_Address parameter of *.*.*.*. This is not allowed.

System Action: OMPROUTE continues, ignoring the specified configuration statement.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file, replacing the full wildcard configuration statement with either explicit statements for each interface or more explicit wildcard statements.

Source Data Set: EZAORYAC

EZZ7894I Neighbors must be coded for non-broadcast capable *stmt name*

Explanation: OMPROUTE found the specified configuration statement in the OMPROUTE configuration file. This statement is for a non-broadcast link. In this situation, neighbors must be coded on the configuration statement. Without this information, OMPROUTE cannot communicate over the link.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file, providing neighbor definitions for the neighbors that can be reached over the non-broadcast link.

Source Data Set: EZAORCFG

EZZ7895I Processing console command - *command*

Explanation: OMPROUTE received the specified DISPLAY or MODIFY command.

System Action: OMPROUTE processes the received command.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMFY

EZZ7896I Could not obtain stack routing table, ioctl errno=*errno*:*description*, **errno2=***errnojr*

Explanation: The attempt to obtain the contents of the TCP/IP stack's route table failed with the specified error.

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

description describes the meaning of the *errno*.

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: EZAORRTI

EZZ7897I User is not RACF authorized to start OMPROUTE

Explanation: The user that attempted to start the OMPROUTE application is not RACF authorized to start it. The user must have RACF authority to the entity MVS.ROUTEMGR.OMPROUTE.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Provide RACF authority to the user as described in the *z/OS Communications Server: IP Configuration Guide*.

Source Data Set: EZAORINI

EZZ7898I OMPROUTE Initialization Complete

Explanation: The OMPROUTE application completed its initialization.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: OMPROUTE

Procedure Name: Store message in syslog. Write message to user log.

EZZ7899I Temporary file allocation failed processing OMPROUTE command

Explanation: A temporary file needed to process the OMPRoute console command could not be allocated.

System Action: The command will not be processed.

User or Operator Response: None.

System Programmer Response: Ensure that space is available in the HFS directory defined in the TMPDIR environment variable or in /tmp if TMPDIR is not defined.

Source Data Set: EZAORMFY

EZZ7900I Bad length packet, from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. The OSPF length field indicates a longer packet than indicated by the IP header length field.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPF

EZZ7901I Bad packet checksum, from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. The packet has an invalid OSPF checksum.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPF

EZZ7902I Bad OSPF version, from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. The version field in the OSPF header is not equal to 2.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPF

EZZ7903I No matching SPF-interface for packet from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. Either the IP destination specified in the packet is not acceptable, or the parameters in the OSPF header (like area ID) do not match the parameters configured for the receiving interface. This may be an acceptable situation if you have a physical network broken into multiple networks using TCP/IP.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7904I Packet authentication failure, from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. The packet fails to authenticate.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Verify the authentication type and authentication key specified for the appropriate interfaces on this and the source router. The types and keys must match in order for authentication to succeed.

Source Data Set: SPF

EZZ7905I No matching OSPF neighbor for packet from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. The packet is not a hello packet, and does not match any existing OSPF neighbor. This is an acceptable situation when OMPROUTE was just started and is receiving non-hello packets that the source router has multicasted onto the network prior to an adjacency being formed with the source router.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7906I Bad packet type received from *source*, **type** *type*

Explanation: An OSPF packet of the specified type was received. The OSPF packet type field is invalid.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPF

EZZ7907I Add interface fails for *interface_name* **- exceeded 254 interfaces**

Explanation: OMPROUTE cannot configure an interface because the maximum of 254 configured routing interfaces was exceeded. This limitation encompasses all interfaces, including real interfaces, VIPAs, and Dynamic VIPAs.

interface_name is the name of the interface that cannot be configured.

System Action: The interface is ignored by OMPROUTE and processing continues. Neither RIP nor OSPF will be active over the interface. The interface, and any destinations reached through it, will not be advertised into any OSPF or RIP autonomous systems to which this router is attached.

User or Operator Response: Contact the system programmer.

System Programmer Response: If you want the interface used by RIP or OSPF, define it before the 254 interface limit is reached.

Source Data Set: EZAORMII

Procedure Name: ezaormii

EZZ7908I Received packet type *type* **from** *source*

Explanation: An OSPF packet of the specified type was received from the specified source.

type is the OSPF packet type as described in RFC 2328. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7909I Sending unicast type *type* **dst** *destination* **net** *net_index* **interface** *name*

Explanation: A unicast OSPF packet of the specified type was sent out the specified interface to the specified IP destination.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7910I Sending multicast, type *type*, **destination** *destination* **net** *net_index* **interface** *name*

Explanation: A multicast OSPF packet of the specified type sent out the specified interface to the specified destination.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7911I Retransmitting packet, type *type*, *source* **->** *destination*

Explanation: A unicast OSPF packet of the specified type is being retransmitted, using the specified source and destination.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPF

EZZ7912I No FSM match, interface *interface*, **state** *state*, **event** *event*

Explanation: The specified event occurred while an interface was in the specified state. This occurrence is not covered by the interface Finite State Machine. This often occurs because of harmless timing windows. For example, a hello timer pops for a neighbor with whom adjacency was already lost.

state is the Interface State Code. The Interface State Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Interface State Codes can be one of the following:

1 Down

2 Backup — Not described in RFC 2328 — When more than one OSPF interface is defined to the same subnet, only one of the interfaces can be primary, meaning that it will be the interface to carry the OSPF protocol traffic between OMPROUTE and the subnet. All other interfaces to the same subnet will be in this state. Failure of the primary interface will result in automatic switching of OSPF traffic to one of the backup interfaces.

4 Loopback

8 Waiting

16 Point-to-point

32 DROther

64 DRBackup

128 DR

event is the Interface Event Code. The Interface Event Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Interface Event Codes can be one of the following:

1 InterfaceUp

2 WaitTimer

3 BackupSeen

4 NeighborChange

5 LoopInd

6 UnloopInd

7 InterfaceDown

8 BackupUp — Not described in RFC 2328 — Lower-level protocols have indicated that the network interface is operational. This interface is one of multiple OSPF interfaces defined to the same subnet and it is not the primary interface (the interface to carry the OSPF protocol traffic between OMPROUTE and the subnet). The interface will transition to Backup state.

System Action: The event is ignored.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: SPFIFC

EZZ7913I State change, interface *interface*, **new state** *state*, **event** *event*

Explanation: The specified event occurred on the specified interface, causing its state to transition to the specified new state.

state is the Interface State Code. The Interface State Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Interface State Codes can be one of the following:

1 Down

2 Backup — Not described in RFC 2328 — When more than one OSPF interface is defined to the same subnet, only one of the interfaces can be primary, meaning that it will be the interface to carry the OSPF protocol traffic between OMPROUTE and the subnet. All other interfaces to the same subnet will be in this state. Failure of the primary interface will result in automatic switching of OSPF traffic to one of the backup interfaces.

4 Loopback

8 Waiting

16 Point-to-point

32 DROther

64 DRBackup

128 DR

event is the Interface Event Code. The Interface Event Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Interface Event Codes can be one of the following:

1 InterfaceUp

2 WaitTimer

3 BackupSeen

4 NeighborChange

5 LoopInd

6 UnloopInd

7 InterfaceDown

8 BackupUp — Not described in RFC 2328 — Lower-level protocols have indicated that the network interface is operational. This interface is one of multiple OSPF interfaces defined to the same subnet and it is not the primary interface (the interface to carry the OSPF protocol traffic between OMPROUTE and the subnet). The interface will transition to Backup state.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFIFC

EZZ7914I No match for hello received on virtual link, from *source*

Explanation: A hello packet was received that could only match a virtual link, yet that link is not configured.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFNBR

EZZ7915I Network mask mismatch in hello from *source*

Explanation: A hello packet was received from the specified neighbor. The neighbor disagrees with this router concerning the network mask of their common network.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Modify the configuration on either the source or local router, ensuring that the network masks configured for the common network match.

Source Data Set: SPFNBR

EZZ7916I Hello interval mismatch in hello from *source*

Explanation: A hello packet was received from the specified neighbor. The neighbor disagrees with this router concerning the hello interval to be used on the common network.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Modify the configuration on either the source or local router, ensuring that the hello intervals configured for the common network match.

Source Data Set: SPFNBR

EZZ7917I Dead interval mismatch in hello from *source*

Explanation: A hello packet was received from the specified neighbor. The neighbor disagrees with this router concerning the ″dead router interval″ to be used on the common network.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Modify the configuration on either the source or local router, ensuring that the dead router intervals configured for the common network match.

Source Data Set: SPFNBR

EZZ7918I No FSM match, neighbor *neighbor***, state** *state***, event** *event*

Explanation: The specified event was generated for the specified neighbor, which is currently in the specified state. This was not anticipated by the neighbor Finite State Machine. Also, this message often occurs after OMPROUTE is stopped and restarted within the dead router interval seconds.

state is the Neighbor State Code. The Neighbor State Codes are described in RFC 2328. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC. The Neighbor State Codes can be one of the following:

1 Down

2 Attempt

4 Init

8 2-way

16 ExStart

32 Exchange

64 Loading

128 Full

event is the Neighbor Event Code. The Neighbor Event Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Neighbor Event Codes can be one of the following:

1 HelloReceived

2 Start

3 2-WayReceived

4 AdjOK?

5 NegotiationDone

6 ExchangeDone

7 SeqNumberMismatch

8 BadLSReq

9 LoadingDone

10 1-way

11 KillNbr

12 InactivityTimer

13 LLDown

14 AdjStart — Not described in RFC2328 — Bidirectional communication was established with the neighbor and this is a neighbor with whom an adjacency should be established.

15 NoProg — Not described in RFC2328 — This is an indication that adjacency establishment with the neighbor failed to complete within a reasonable time period (Dead_Router_Interval seconds). Adjacency establishment restarts.

System Action: The event is ignored.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: SPFNBR

EZZ7919I State change, neighbor *neighbor*, **new state** *state*, **event** *event*

Explanation: The specified event was generated, causing the specified neighbor to transition to the specified new state.

state is the Neighbor State Code. The Neighbor State Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Neighbor State Codes can be one of the following:

1 Down

2 Attempt

4 Init

8 2-way

16 ExStart

32 Exchange

64 Loading

128 Full

event is the Neighbor Event Code. The Neighbor Event Codes are described in RFC 2328. See Appendix B, "Related protocol specifications (RFCs)" on page 733 for directions to get a copy of the RFC. The Neighbor Event Codes can be one of the following:

1 HelloReceived

2 Start

3 2-WayReceived

4 AdjOK?

5 NegotiationDone

6 ExchangeDone

7 SeqNumberMismatch

8 BadLSReq

9 LoadingDone

10 1-way

11 KillNbr

12 InactivityTimer

13 LLDown

14 AdjStart — Not described in RFC2328 — Bidirectional communication was established with the neighbor and this is a neighbor with whom an adjacency should be established.

15 NoProg — Not described in RFC2328 — This is an indication that adjacency establishment with the neighbor failed to complete within a reasonable time period (Dead_Router_Interval seconds). Adjacency establishment restarts.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFNBR

EZZ7920I Outstanding Database Description packet not avail for neighbor *neighbor*

Explanation: An attempt was made to retransmit a Database Description packet to the specified neighbor, but the packet could not be found.

System Action: Retransmission is aborted.

User or Operator Response: None.

System Programmer Response: Possible internal error. Contact the IBM Software Support Center.

Source Data Set: DBXCHG

EZZ7921I OSPF Adjacency Failure, neighbor *neighbor*, **old state** *state*, **new state** *state*, **event** *event*

Explanation: The specified event was generated, causing the specified OSPF adjacency to transition to the specified new state.

state is the Neighbor State Code. The Neighbor State Codes are described in RFC 1583. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC. The Neighbor State Codes can be one of the following:

1 Down

2 Attempt

4 Init

8 2-way

16 ExStart

32 Exchange

64 Loading

128 Full

event is the Neighbor Event Code. The Neighbor Event Codes are described in RFC 1583. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC. The Neighbor Event Codes can be one of the following:

7 SeqNumberMismatch

8 BadLSReq

10 1-way

11 KillNbr

12 InactivityTimer

13 LLDown

15 NoProg — This code is not described in RFC1583. This is an indication that adjacency establishment with the neighbor failed to complete within a reasonable time period (Dead_Router_Interval seconds). Adjacency establishment restarts.

System Action: None.

User or Operator Response: If connectivity problems occur investigate source of adjacency establishment failure.

System Programmer Response: None.

Source Data Set: nbr_fsm()

Procedure Name: SPFNBR

EZZ7922I Bad length Link state advertisement received from *neighbor*

Explanation: A link state advertisement was received from the specified neighbor, and the advertisement’s length field indicates that the entire advertisement is NOT fully contained in the received Link State Update Packet.

System Action: The partial advertisement is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPFLOOD

EZZ7923I from *neighbor*, **LS advertisement checksum fails: LS type** *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was received from the specified neighbor. The advertisement is identified by its LS type and two-part originating ID. The checksum field contained in the advertisement is invalid.

System Action: The advertisement is ignored.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPFLOOD

EZZ7924I from *neighbor*, **bad LS type, advertisement: typ** *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was received from the specified neighbor. The advertisement is identified by its LS type and two-part originating ID. The advertisement’s LS type field is invalid.

System Action: The advertisement is ignored.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPFLOOD

EZZ7925I from *neighbor*, **AS external link adv. on Virtual Link: typ** *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was received from the specified neighbor. It was received over a virtual link, yet its LS type is equal to AS external link. AS external link state advertisements should not be sent over virtual links.

System Action: The advertisement is ignored.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPFLOOD

EZZ7926I from *neighbor*, **old LS advertisement: typ** *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was received from the specified neighbor. The advertisement is older than the current database copy.

System Action: The advertisement is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7927I from *neighbor*, **self update: typ** *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was received. The advertisement was originated by the router itself, yet is newer than the database copy. This indicates that it originated before the router was last started. This causes the router to either advance the LS sequence number and originate a new instance of the advertisement, or delete the advertisement, if it is a summary LSA and the attached area does not wish to import summary LSAs anymore.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7928I from *neighbor*, **new LS advertisement: typ** *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was received from the specified neighbor. The advertisement is newer than the current database copy.

System Action: The advertisement is flooded out all other interfaces, and installed in the routing database.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7929I from *neighbor*, **Old acknowledgment for advertisement: typ** *type* **id** *destination* **org** *source*

Explanation: An unexpected link state acknowledgment was received from the specified neighbor. The acknowledgment, however, is for a previous instance of the link state advertisement.

System Action: The acknowledgment is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFACK

EZZ7930I Bad acknowledgment from *neighbor* **for advertisement: typ** *type* **id** *destination* **org** *source*

Explanation: An unexpected link state acknowledgment was received from the specified neighbor. The acknowledgment, however, is for the current instance of the link state advertisement.

System Action: The acknowledgment is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFACK

EZZ7931I LS update retransmission to neighbor *neighbor*

Explanation: A Link State Update packet containing retransmitted link state advertisements was unicast to the specified neighbor. This probably indicates packet loss during the flooding procedure.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFACK, SPFLOOD

EZZ7932I LS acknowledgment sent directly to neighbor *neighbor*

Explanation: A Link State Acknowledgment packet was sent directly to the specified neighbor. This is in response to duplicate link state advertisements received from the neighbor. This probably indicates packet loss during the flooding procedure.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7933I Flushing advertisement: typ *type* **id** *destination* **org** *source*

Explanation: The specified link state advertisement contained in the link state database was not refreshed for 2 hours. The advertisement is deleted from the database. This probably indicates that the originator of the advertisement is unreachable.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFPARSE, SPFTIMER

EZZ7934I Originating LS advertisement: typ *type* **id** *destination* **org** *source*

Explanation: The specified link state advertisement is being (re)originated by the router. This can be due to topological change, or the necessity to refresh.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFORIG

EZZ7935I New OMPROUTE route to destination *desttype destination*, **type** *routetype* **cost** *cost*

Explanation: The OSPF routing table build process detected a new best route to the specified destination, having the specified cost.

System Action: The new route will be placed in OMPROUTE's internal route table, replacing any existing route to the same destination if the new route is better. The new route will also be added to the TCP/IP stack's route table providing that there is not a static route to the destination defined to the TCP/IP stack using the GATEWAY configuration statement.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFRTTBL

EZZ7936I Unicast hello sent to IP destination *neighbor*

Explanation: An OSPF hello was sent to the specified IP destination. The hello packet was sent using unicast.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NBMA

EZZ7937I The OSPF routing protocol is *disposition*

Explanation: Displayed on router startup. Indicates operational status of the OSPF protocol.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORYAC, SPFCFGCH, SPFCONF

EZZ7938I SPF Interface *interface* (*name*) **is not an IP interface, interface not installed**

Explanation: Displayed on router startup when an OSPF interface address is configured, yet this address was not configured to the TCP/IP stack.

System Action: The OSPF interface is not installed.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file or the TCP/IP stack configuration, ensuring that the specified interface is configured in both.

Source Data Set: SPFCONF

EZZ7939I Duplicate LS acknowledgment received from neighbor *neighbor*

Explanation: Unexpected link state acknowledgment was received from the specified neighbor. This probably indicates packet loss during the flooding procedure.

System Action: The link state acknowledgment is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFACK

EZZ7940I from *neighbor*, **bad age field, advertisement: typ** *type* **id** *destination* **org** *source*

Explanation: The specified link state advertisement was received from the specified neighbor. The advertisement's LS age field is invalid.

System Action: The advertisement is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7941I Transit area *area* **not configured, virtual link ignored**

Explanation: A virtual link was configured to have a certain transit area, yet that area was not configured.

System Action: The virtual link is ignored.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file to correct the transit area configured for the virtual link, if incorrect, or to configure the area.

Source Data Set: SPFCONF

EZZ7942I Backbone area is not configured, all virtual links discarded

Explanation: OMPROUTE found Virtual_Link configuration statements in the OMPROUTE configuration file, but the backbone area is not configured. Virtual links cannot be used unless a backbone area is configured.

System Action: All Virtual_Link configuration statements are ignored.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration statement to configure the backbone area.

Source Data Set: SPFCONF

EZZ7943I Destination *desttype destination* **now unreachable**

Explanation: The specified destination was found to be unreachable during the OSPF routing table build process.

System Action: The route to the specified destination will be removed from OMPROUTE's internal route table as well as the TCP/IP stack's route table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFRTTBL

EZZ7945I from *neighbor*, **received unexpected MaxAge: typ** *type* **id** *destination* **org** *source*

Explanation: The specified link state advertisement was received from the specified neighbor. Its age is MaxAge and there is no current instance of the advertisement in the router's database.

System Action: The advertisement is acknowledged and then discarded without flooding.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7946I error in advertisement: typ *type* **id** *destination* **org** *source*

Explanation: The specified link state advertisement was received from the specified neighbor. The advertisement contains an error.

System Action: The advertisement is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPFLOOD, SPFPARSE

EZZ7947I Stub area mismatch in hello from *source*

Explanation: A hello packet was received from the specified neighbor. The neighbor disagrees with this router concerning the attached area's ability to process AS external link advertisements.

System Action: The Hello packet is ignored.

User or Operator Response: None.

System Programmer Response: Modify the configuration on either the source or local router, ensuring that the routers agree on the attached area's stub status.

Source Data Set: SPFNBR

EZZ7948I from *neighbor***, type 5 LSA in stub area, adv: typ** *type* **id** *destination* **org** *source*

Explanation: A type 5 (AS External Link) link state advertisement was received from the specified neighbor. The advertisement is being flooded through a stub area, and is therefore ignored.

System Action: The link state advertisement is discarded.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the OSPF application on the source router to report the problem.

Source Data Set: SPFLOOD

EZZ7949I Dijkstra calculation performed, on *number* **area(s)**

Explanation: As a result of a topology change, the routing table was recalculated, starting with the Dijkstra calculation.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFRTTBL

EZZ7950I Network LSA with old Advertising Router: (*type***,***destination***,***source***)**

Explanation: A network links advertisement having one of our addresses as Link State ID, but whose Advertising Router is not our Router ID, was received. These advertisements are flushed, as they are assumed to be out-of-date.

System Action: The advertisement is flushed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7951I Reparsing Network LSA: *LS_ID*

Explanation: A network link is being reparsed, owing to the fact that there are multiple network-LSAs in the network with the same Link State ID. This indicates that a router changed OSPF Router IDs, and originated the same router-LSA before and after the change. This is a normal, but rare, event.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFPARSE

EZZ7956I OSPF area *area* **not configured, interface** *interface* **not installed**

Explanation: Displayed on router startup when an OSPF interface address is configured, but the attached area configured for the interface is not a configured area.

System Action: OMPROUTE continues, but the specified interface will not be used as an OSPF interface.

User or Operator Response: None.

System Programmer Response: Modify the OMPROUTE configuration file to correct the Attached_Area parameter on the specified OSPF_Interface statement, if incorrect, or to configure the area.

Source Data Set: SPFCONF

EZZ7961I Demand circuit support active for area *area*

Explanation: Displayed when there are no more DC bit clear LSAs in any of the area's link state databases and it is valid to set the DoNotAge bit.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFUTIL

EZZ7962I Demand circuit support not active for area *area*

Explanation: Displayed when an LSA with the DC bit clear is added to one of the area's link state databases and any LSAs with the DoNotAge bit set are purged.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFUTIL

EZZ7963I Unchanged advertisement: typ *type* **id** *destination* **org** *source* **suppressed for demand interfaces**

Explanation: Displayed when an LSA is not flooded over one or more circuits configured as demand circuits because there is no change in the content of the LSA from a previous version.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFLOOD

EZZ7964I Hello's on interface *interface* **to neighbor** *neighbor* **are being suppressed.**

Explanation: Displayed when hello suppression becomes active for the specified interface and neighbor.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFNBR

EZZ7965I Cbit clear indicate LSA received in area *area* **from** *source***.**

Explanation: Displayed when a special type 4 indicate LSA is received in a non stub area to indicate the presence of routers outside the area that do not support DoNotAge processing.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFPARSE

EZZ7966I Cbit clear indicate LSA originated in area *area***.**

Explanation: Displayed when the local router originates a special type 4 indicate LSA to indicate the presence of routers outside the area that do not support DoNotAge processing.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SPFORIG

EZZ7967I advertisement discarded, overflows buffer: LS type *type* **id** *destination* **org** *source*

Explanation: A link state advertisement was discarded because it would be too large to fit in the router's data area. A router links LSA became excessively large due to a large number of direct connections to neighbor routers within a single area.

System Action: None.

User or Operator Response: None.

System Programmer Response: Reconfigure the network to reduce the size of the largest link state advertisement or reconfigure the router to increase the size of router data areas to hold the LSA. The size of the router data areas to hold LSAs is equal to the largest of maximum MTU sizes defined on OSPF_INTERFACE configuration statements. The data area used to build LSAs can be increased to the size of the local router's buffer by configuring the maximum LSA size. The size of the local router's buffer can be enlarged by increasing the size of the largest MTU for a locally attached subnetwork.

Source Data Set: SPFLOOD, SPFORIG

EZZ7974I Could not open /dev/null for writing.

Explanation: OMPROUTE opens /dev/null for writing to avoid the display of duplicate OMPROUTE error messages on the system console. The attempt to open /dev/null for writing failed.

System Action: OMPROUTE ends.

User or Operator Response: None.

System Programmer Response: Ensure that /dev is in a write-enabled HFS and ensure that the file permissions for /dev/null allow for writing by OMPROUTE.

Source Data Set: main

Procedure Name: OMPROUTE

EZZ7968I HELLO INTERVAL MISSED ON INTERFACE *interface*

Explanation: This message indicates that at least one interface missed hello intervals and the Dead Router Interval is approaching. A routing daemon for a complex network topology has many time-sensitive tasks. On busy systems, if these tasks exceed a certain time period (Dead Router Interval), adjacent routers will determine that OMPROUTE became inactive and drop routes.

interface is the name of the OSPF interface that missed at least two hello intervals.

System Action: OMPROUTE continues.

User or Operator Response: Verify that the interface is active on the network.

System Programmer Response: OMPROUTE might not be configured with the proper dispatching priority to process the dynamic routing protocol traffic for the network. If adjacent routers drop routes learned from OMPROUTE, then the dispatching priority of OMPROUTE must be increased or the Dead Router Interval lengthened.

Source Data Set: unnecessary

Procedure Name: SPFTIMER

Chapter 8. EZZ8*xxxx* messages

EZZ8000I bad version *version* **received from host** *source*

Explanation: The version field in the RIP header did not match the current version. This is probably caused by an error in the source host.

System Action: The received RIP packet is ignored.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the source host and report the problem.

Source Data Set: INRIP

EZZ8001I request received from host *source*

Explanation: A RIP routing table request was received from another host.

System Action: A routing table update will be sent to the requesting host.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIP

EZZ8002I trace on to *file* **received from host** *source*

Explanation: A request from a host to turn RIP tracing on to a given log file was received.

System Action: This request is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIP

EZZ8003I trace off received from host *source*

Explanation: A request from a host to turn RIP tracing off was received.

System Action: This request is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIP

EZZ8004I response received from host *source*

Explanation: A RIP routing table update was received. Note that it may take more than one response packet to transmit the entire routing table, especially if the routing table is large.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIP

© Copyright IBM Corp. 1994, 2002

EZZ8005I bad command code *command* **received from host** *source*

Explanation: A RIP message was received with an unrecognized command code. This is probably caused by an error or out of date software in the source host.

System Action: The received RIP packet is ignored.

User or Operator Response: None.

System Programmer Response: Contact the manufacturer of the source host and report the problem.

Source Data Set: INRIP

EZZ8006I response received from off network host *source*

Explanation: A RIP routing update response was received from a machine which was not directly attached to the network the response came in on. Since normal RIP software is generally written to send data only to connected nets, this is probably indicative of a hostile event.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: Examine audit trails and other information to determine the original source host.

Source Data Set: INRIPIN

EZZ8007I Global RIP filters must be type NOSEND or NORECEIVE

Explanation: Only filter types of NOSEND or NORECEIVE are valid for a global RIP filter.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Correct the value coded for the specified keyword in the OMPROUTE configuration file.

EZZ8008I dynamic route to *destination* **from** *source* **disallowed**

Explanation: A dynamic route was received but is being ignored because the configuration of RIP on the router does not allow dynamic routes except for those in a table, and this route was not in that table.

System Action: Received dynamic route is ignored.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPIN

EZZ8009I network *destination* **now unreachable via router** *router***, deleted**

Explanation: An incoming RIP update from the router that was previously listed as the next hop to the destination network announced that the destination is unreachable (at metric 'infinity'). The RIP route to that destination is being deleted.

System Action: The route is deleted from OMPROUTE's internal route table as well as the TCP/IP stack's route table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPIN

EZZ8010I update route to net *destination* **at metric** *metric* **hops via router** *router*

Explanation: A new (better) route to the given destination was learned using RIP and was installed.

System Action: The new route will be placed in OMPROUTE's internal route table, replacing any existing route to the same destination if the new route is better. The new route will also be added to the TCP/IP stack's route table providing that there is not a static route to the destination defined to the TCP/IP stack using the GATEWAY configuration statement.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPIN

EZZ8011I send request to address *source*

Explanation: OMPROUTE is sending a RIP request from each of the addresses associated with an interface that just came up.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPOUT

EZZ8012I sending broadcast response to address *destination* **in** *count* **packets with** *number* **routes**

Explanation: The router is sending a normal RIP broadcast update (triggered either by a timer or a change in the routing table) to the specified address.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPOUT

EZZ8013I sending response to address *destination* **in** *count* **packets with** *number* **routes**

Explanation: The router is sending a RIP update (triggered by a request from another host) to the specified address.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPOUT

EZZ8014I Blackhole RIP filters must be type NOSEND or NORECEIVE

Explanation: Only filter types of NOSEND or NORECEIVE are valid when * is coded as the destination network.

System Action: OMPROUTE terminates.

User or Operator Response: None.

System Programmer Response: Correct the value coded for the specified keyword in the OMPROUTE configuration file.

Source Data Set: EZAORYAC

EZZ8015I sending packet to *destination*

Explanation: A RIP packet (either a routing table update, or when an interface first comes up, a request) was sent to the specified destination.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPOUT

EZZ8016I Global RIP Filters

Explanation: This message precedes the display of the global RIP filters in response to the D TCPIP,proc,OMPROUTE,RIP,FILTERS command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRPCON

EZZ8017I network route to *destination* **timed out**

Explanation: There was a route in the route table to the specified destination by way of a router that was not heard from for a while. The route was deleted.

System Action: The route is marked unreachable in OMPROUTE's internal route table and is deleted from the TCP/IP stack's route table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPTIM

EZZ8018I network route to *destination* **deleted**

Explanation: There was a route in the route table to the specified destination by way of a router that was not heard from for a while. The route was previously marked unreachable, and is now being deleted.

System Action: The route is deleted from OMPROUTE's internal route table.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPTIM

EZZ8019I Mismatch version *version* **received from host** *source*

Explanation: The version field in the received RIP header did not match the configured version on the receiving interface.

System Action: The received RIP packet is ignored.

User or Operator Response: None.

System Programmer Response: Modify the configuration on either the source or local router, ensuring that the RIP version configured for the appropriate interfaces match.

Source Data Set: INRIP

EZZ8020I Authentication error received from host *source*

Explanation: A RIP packet from the specified host was rejected due to an authentication error caused by either invalid authentication info or authentication not being enabled. This is probably caused by a misconfiguration.

System Action: None.

User or Operator Response: None.

System Programmer Response: Modify the configuration on either the source or local router, ensuring that the authentication configurations are consistent.

Source Data Set: INRIP

EZZ8021I sending RIP2 response to address *destination* **from** *source* **in** *count* **packets with** *number* **routes**

Explanation: The router is sending a normal RIP2 update (triggered either by a timer or a change in the routing table) to the specified address.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPOUT

EZZ8022I Destination *dst* **filtered on** *dir* **RIP broadcast (***name***)**

Explanation: A destination on an inbound or outbound RIP broadcast was filtered due to a FILTER statement.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRIPIN, INRIPOUT

EZZ8023I The RIP routing protocol is *disposition*

Explanation: Displayed on router startup. Indicates operational status of the RIP protocol.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: OMPROUTE

EZZ8024I RIP message received on non RIP interface *source* **from** *destination* **is ignored.**

Explanation: A RIP routing message was received on an interface which is not configured as a RIP interface.

System Action: The packet is discarded.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORRRT

EZZ8025I RIP send to *destination* **fails**

Explanation: An attempt to send a RIP packet to the indicated destination failed. This may be a temporary condition, however if the message is issued repeatedly this condition requires attention.

System Action: None.

User or Operator Response: None.

System Programmer Response: Other OMPROUTE messages will give the errno explanation for this failure.

Source Data Set: INRIPOUT

EZZ8026I RIP packet from router *router–name* **ignored**

Explanation: OMPRoute was configured to ignore RIP packets from *router–name*.

System Action: OMPRoute continues.

User or Operator Response: If it is desired to use RIP packets from the router specified, update the OMPRoute profile to accept RIP packets from this router.

System Programmer Response: none

Source Data Set: INRIP

Procedure Name: rip_in

EZZ8050I Updating BSD Route Parms for link *linkname*, **MTU** *mtuval*, **metric** *metricval*, **subnet** *subnetval*, **destination** *destval*

Explanation: OMPROUTE is updating the stack's BSD Routing Parameters for the specified link so that it will agree with the data being used by OMPROUTE.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ8054I Add failed for net *network*; **bad network number**

Explanation: This message is generated when a network cannot be added to the routing table because of a bad network number. This software considers the net above to be invalid.

System Action: None.

User or Operator Response: None.

System Programmer Response: If the net is valid, there is a possible internal error, contact the IBM Software Support Center .

Source Data Set: INRTE

EZZ8055I Re-adding static route to net *network*

Explanation: This message is generated when a static route to a network is brought back into use.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INRTE

EZZ8057I Added network *network* **to interface** *interface* **on net** *net_index* **interface** *name*

Explanation: This message is generated when a route to a directly-connected network is added to OMPROUTE's internal routing table as a result of the route being defined to the TCP/IP stack using the GATEWAY statement.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTBL

EZZ8058I Ignoring bad static route to *network*, **mask** *mask*

Explanation: This message is generated when a bad static route is encountered.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INSTATIC

EZZ8059I Added network *network* **with route via** *gateway* **on net** *net_index* **interface** *name*

Explanation: This message is generated when a route to a indirectly-connected network is added to OMPROUTE's internal routing table as a result of the route being defined to the TCP/IP stack using the GATEWAY statement.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INSTATIC

EZZ8061I Deleted net *network* **route via** *gateway* **net** *net_index* **interface** *name*

Explanation: This message is generated when an interface goes down and the corresponding route is deleted from the route table.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INUPDWN

EZZ8062I Subnet *network* **defined**

Explanation: This message is generated when a new subnetted network is defined.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTBL

EZZ8063I Deleting subnetted network *network*

Explanation: This message is generated when a subnetted network is deleted. This happens when there are no longer any interfaces to that network.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTBL

EZZ8064I RIP disabled on *interface* **variable length subnet masks**

Explanation: The router is configured with variable length subnet masks on the same network, which RIP cannot handle. Thus RIP is disabled on the interface.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INCONFRP

EZZ8066I IP protocol does not run over *type*

Explanation: An IP address was configured for a type of network which currently doesn't support IP.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INCONF

EZZ8067I Network *net_index* **interface** *name* **is inactive**

Explanation: This message is generated when an inactive network interface is encountered during initialization, no direct routes will be created on this network.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: OMPROUTE

EZZ8072I Received *type* **logical connection over** *name*, **destination** *dest*

Explanation: OMPROUTE learned of a status change of the specified type for the logical connection to the specified destination over the specified TCP/IP interface.

System Action: OMPROUTE makes any necessary changes to its processing based upon the status change. Also, any necessary changes are made to routes that use this logical connection.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMII

EZZ8073I Dynamically added *intf name* **to omproute configuration**

Explanation: A new OSPF_Interface, RIP_Interface, or Interface statement was successfully read from the OMPRoute configuration file as a result of issuing F proc,RECONFIG

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORYAC

EZZ8074I OMPROUTE processing error

Explanation: OMPROUTE experienced a processing error.

System Action: OMPROUTE terminates. This message is followed by an EZZ7805I message that provides the processing return code.

User or Operator Response: Contact your system programmer.

System Programmer Response: Rectify the problem indicated by the return code given on the EZZ7805I message that follows this one.

Source Data Set: OMPROUTE

EZZ8075I Retrying queued stack route updates

Explanation: Stack route update attempts which earlier failed and were queued for retry are now being retried. Retries that fail will not be retried again.

System Action: OMPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ8076I Stack route update retries complete

Explanation: All stack route updates which were waiting to be retried have been attempted.

System Action: OMPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORMUP

EZZ8077I Ignoring replaceable static route to *dest*, **mask** *mask* **using** *gateway* **-** *reason*

Explanation: TCP/IP informed OMPROUTE of a new replaceable static route to *dest* with *mask*, using *gateway*. OMPROUTE did not accept the route from TCP/IP for *reason*.

dest is the destination of the route described in the ignored replaceable static route.

mask is the subnet mask of the destination of the route described in the ignored replaceable static route.

gateway is the gateway used by the replaceable static route to reach the destination.

reason is the reason OMPROUTE ignored the replaceable static route. At this time, the only value for *reason* is:

dynamic routes already active
OMPROUTE is already aware of dynamic routes to the destination, and dynamic routes are preferable to replaceable static routes.

System Action: Processing continues, the existing routes to dest are not modified.

User or Operator Response: Contact the system programmer.

System Programmer Response: If the intention was for the static route to override dynamic routes, do not define it as REPLACEABLE.

Source Data Set: EZAORMII

Procedure Name: ezaormii

EZZ8100I OMPROUTE subagent Starting

Explanation: The OMPROUTE subagent is starting to initialize.

System Action: The OMPROUTE subagent is attempting to connect to the SNMP agent.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: EZAORDPI

EZZ8101I OMPROUTE subagent Initialization Complete

Explanation: The OMPROUTE subagent completed initialization and is ready to start processing.

System Action: The OMPROUTE subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None

Source Data Set: EZAORDPI

EZZ8102I OMPROUTE subagent Internal Error *code*

Explanation: Internal Error occurred check the following codes:

Code	Meaning
01	mkDPIopen failed
02	pDPIpacket failed
03	no DPI response to DPI OPEN packet
04	SNMP agent rejected the DPI OPEN request
05	OMPROUTE subagent not authorized to agent
06	DPIget_fd_for_handle failed
07	mkDPIregister failed
08	mkDPIresponse failed
09	Severe processing error
10	mkDPIset failed

System Action: The OMPROUTE subagent is disconnected from the SNMP agent.

User or Operator Response: Re-create the problem with the SNMP agent -d trace option and OMPROUTE -s 1 trace option and make sure that Syslog Daemom (syslogd) is running. Contact the TCP/IP administrator.

System Programmer Response: Error information will be written to the syslogd output file. Contact your IBM Software Support Center with the trace information from the SYSLOGD.

Source Data Set: EZAORDPI

EZZ8103I OMPROUTE subagent Shutdown Complete

Explanation: OMPROUTE subagent disconnected from the SNMP agent.

System Action: The OMPROUTE subagent is disconnected from the SNMP agent.

User or Operator Response: Restart OMPROUTE subagent.

System Programmer Response: None

Source Data Set: EZAORDPI

EZZ8104I OMPROUTE subagent: duplicate subagent identifier error

Explanation: The SNMP agent rejected the DPI OPEN request from the OMPROUTE subagent because another subagent was already connected to the SNMP agent using the same subagent identifier.

System Action: The OMPROUTE subagent is disconnected from the SNMP agent.

User or Operator Response: Ensure that no user DPI programs are using this 1.3.6.1.4.1.2.11.7.3 subagent identifier.

System Programmer Response: None.

Source Data Set: EZAORDPI

EZZ8105I OMPROUTE subagent: waiting for group *MIB_Object*

Explanation: The OMPROUTE subagent attempted to register the specified MIB Object, *MIB_Object*. One of the following could have occurred:

- Another DPI subagent had already registered the requested MIB Object with a higher priority.
- The OMPROUTE subagent had successfully registered the specified MIB Object, but the SNMP agent received a later registration from another subagent which requested a higher priority than that with which the OMPROUTE subagent was currently registered.

System Action: The OMPROUTE subagent will continue processing for the other MIB Objects supported by the

OMPROUTE subagent. If the MIB Object becomes available at a later time the OMPROUTE subagent will begin processing for that MIB Object.

User or Operator Response: If it is acceptable that a DPI subagent program other than the OMPROUTE subagent provide the processing for the MIB variables in the specified MIB Object, then no action is necessary. Otherwise, the other DPI subagent must be ended in order for the OMPROUTE subagent to provide the processing for the variables in the specified MIB Object.

System Programmer Response: None.

Source Data Set: EZAORDPI

EZZ8106I OMPROUTE subagent: group *MIB_Object* unregistered by Manager

Explanation: The OMPROUTE subagent received a notification that the MIB Object *MIB_Object* was unregistered. This action was initiated by a request from an SNMP manager.

System Action: The OMPROUTE subagent will continue processing for the other MIB Object supported by the OMPROUTE subagent. The variables under the MIB Object unregistered by the SNMP Manager will no longer be available.

User or Operator Response: If it is acceptable that the specified MIB Object is no longer available, then no action is necessary. To regain the unregistered MIB Object, the OMPROUTE subagent must be closed by an SNMP Manager. Do this by setting the saStatus for the OMPROUTE subagent to not valid (2). This will cause the OMPROUTE subagent to disconnect from the SNMP agent, then reconnect and reregister all of its supported MIB Objects, including any which were previously unregistered by an SNMP Manager request.

System Programmer Response: None.

Source Data Set: EZAORDPI

EZZ8107I OMPROUTE subagent: Connection to SNMP agent Dropped

Explanation: The OMPROUTE subagent was connected to the SNMP agent, but the connection was broken.

System Action: The OMPROUTE subagent will try to reconnect to the SNMP agent until reconnected.

User or Operator Response: If the SNMP agent job is not active, restart the SNMP agent. If the SNMP agent is currently active, the OMPROUTE subagent should automatically reconnect to the agent. If this does not occur, then stop the SNMP agent and restart it.

System Programmer Response: None.

Source Data Set: EZAORDPI

EZZ8108I OMPROUTE subagent: reconnected to SNMP agent

Explanation: The OMPROUTE subagent reconnected to the SNMP agent after detecting that the prior connection had been broken.

System Action: The OMPROUTE subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORDPI

EZZ8109I OMPROUTE subagent: Connecting to loopback address to SNMP agent

Explanation: The OMPROUTE subagent was unable to resolve the local host address and is using the loopback address to connect to the SNMP agent instead of the host address.

System Action: The OMPROUTE subagent will try to connect to the agent using the loopback address.

User or Operator Response: Contact the System Programmer.

System Programmer Response: The following should be considered:

- Ensure that default home internet address is valid in the PROFILE.TCPIP data set.

- If the loopback address is used to connect to the SNMP agent, and a password other than the SNMP agent's -c default password is used by the OMPROUTE subagent when connecting, then the password used by the OMPROUTE subagent must be defined for the loopback address 127.0.0.1 in the SNMP agent's configuration data file.
- If possible, correct the indicated error. If necessary, contact your IBM Software Support Center with the trace output.

Source Data Set: EZAORDPI

EZZ8110I Unable to start OMPROUTE subagent

Explanation: A MODIFY ROUTESA command was entered to enable the OMPROUTE subagent, but the subagent was unable to be started.

System Action: OMPROUTE processing continues without the OMPROUTE subagent.

User or Operator Response: Re-create the problem with the OMPROUTE -d3 trace option. Contact the TCP/IP administrator.

System Programmer Response: Error information will be written to the debug output destination file. This defaults to stdout, but can be written to a file by using the OMPROUTE_DEBUG_FILE environment variable before starting OMPROUTE. Contact your IBM Software Support Center with the resulting debug information.

Source Data Set: EZAORRTI

EZZ8111I OMPROUTE subagent is already *disposition*

Explanation: A MODIFY ROUTESA command was entered to enable or disable the OMPROUTE subagent, but the subagent was already enabled or disabled.

System Action: The OMPROUTE subagent remains enabled or disabled.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZAORRTI

EZZ8201I Pagtsnmp subagent: Internal Error *code*

Explanation: The subagent encountered an internal programming error. The error is described by *code*.

CODE	Meaning
01	mkDPIopen failed
02	failure parsing dpi packet (DPIopen)
03	no DPI response to DPI open
04	agent rejected the Open request - reason unknown
05	subagent not authorized to agent
06	DPIget_fd_for_handle failed
07	failure during mkDPIregister
08	failure parsing DPI packet (DPIregister)
09	Severe error processing packet.

System Action: The pagtsnmp subagent stops.

User or Operator Response: For error code **05** verify that the community name specified for the subagent matches one supported by the SNMP Agent. Otherwise, if possible, re-create the problem with subagent trace level 2. Contact the system programmer.

System Programmer Response: Error information will be written to the Syslog Daemon (syslogd) output file by default. Contact your IBM support center with the syslogd output. Note: Use of the -o startup option sends all debug information to stdout.

Source Data Set: PAGTSNMP

Procedure Name: doConnectAndOpen, doRegister, callFunction

EZZ8202I Pagtsnmp subagent: duplicate subagent identifier error

Explanation: The SNMP Agent rejected the DPI open request from the subagent because another subagent was already connected to the Agent using the same subagent identifier.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Contact your system programmer.

System Programmer Response: The pagtsnmp subagent uses 1.3.6.1.4.1.2.11.7.4 as the subagent identifier. Ensure that no user DPI programs are using this subagent identifier and that no pagtsnmp subagent instance was already started.

Source Data Set: PAGTSNMP

Procedure Name: doConnectAndOpen

EZZ8203I PAGTSNMP SUBAGENT: INITIALIZATION COMPLETE

Explanation: The pagtsnmp subagent completed initialization and is ready to start processing requests.

System Action: The pagtsnmp subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8204I Pagtsnmp subagent: Value specified for the *parm* parameter is missing

Explanation: A value was not entered for the indicated pagtsnmp subagent startup option *parm*.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Fix the parameter specification and restart the pagtsnmp subagent. Refer to *z/OS Communications Server: IP Configuration Reference* for a description of the valid startup options.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: parseParms

EZZ8205I Pagtsnmp subagent: sigaction() failed for *signal* - *errno*

Explanation: The pagtsnmp subagent encountered an error attempting to set up a signal handler.

errno is the Unix System Services Return Code returned by the C Run-time Library for the failing sigaction() call. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*. If the signal handler is not correctly enabled, the subagent will continue processing, but certain functions controlled by the failing signal will not function properly.

signal specifies the particular signal name. 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 may not be properly cleaned up.

SIGTERM
: Handler controls cleanup of resources during termination.

SIGPIPE

Handler allows the subagent to detect when the connection to the SNMP agent was terminated by the agent (for example, if the agent times out while waiting for the subagent response and closes the connection). If sigaction() fails for SIGPIPE, the subagent may either hang or stop.

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

User or Operator Response: None.

System Programmer Response: Contact your IBM support center. Provide the failing *errno* and any error information written to syslogd.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8206I Pagtsnmp subagent: Parameter *parm* not valid

Explanation: The startup option *parm* entered for the pagtsnmp subagent was not valid.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Restart the pagtsnmp subagent with valid parameters. Refer to *z/OS Communications Server: IP Configuration Reference* for a description of the valid startup options.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: parseParms

EZZ8207I Pagtsnmp subagent: unable to open UDP socket to TCPIP - *errno errnojr*

Explanation: The pagtsnmp subagent must open a UDP socket to TCP/IP in order to retrieve the pagtsnmp variable information from TCP/IP. The subagent's attempt to open the UDP socket failed due to the specified error described by *errno* and *errnojr*.

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

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the problem indicated by *errno*.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8208I PAGTSNMP SUBAGENT: PAGTSNMP WAS CANCELLED

Explanation: The pagtsnmp subagent was stopped by either user or operator action.

System Action: The pagtsnmp subagent stops.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: termhand

EZZ8209I Pagtsnmp subagent: *ioctl* **failed with error** *errno*

Explanation: The subagent issued the SIOCGIBMOPT ioctl for the *ioctl* subtype. The ioctl call failed with the specified error *errno*.

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

System Action: The subagent is unable to process the request for any MIB object that requires the associated ioctl to provide its data. An error response is returned to the Agent.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the indicated error. Refer to *z/OS UNIX System Services Messages and Codes* for more information about the socket errors.

Source Data Set: PAGTSMIB

Procedure Name: getIoctl called from getPolicyStatsTable for ioctl subtype SIOCGSLAPTBL.

EZZ8210I PAGTSNMP SUBAGENT: ABEND DETECTED, PAGTSNMP ENDING

Explanation: This message indicates that an internal programming error caused pagtsnmp subagent to abnormally terminate.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Collect debug information via the pagtsnmp -d 2 parameter and forward the results to your system programmer for resolution.

System Programmer Response: Contact the IBM support center with the collected debug information.

Source Data Set: PAGTSNMP

Procedure Name: abndhand

EZZ8211I Pagtsnmp subagent: unable to open message catalog pagtsmsg.cat - *errno*

Explanation: The subagent attempted to open the subagent message catalog pagtsmsg.cat in the message catalog directory, but was unable to open the catalog due to the indicated *errno*. The subagent message catalog should have been installed in (or have a symbolic link provided in) the /usr/lib/nls/msg/C message catalog directory.

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

System Action: The subagent will use the internal default messages instead of the external message catalog.

User or Operator Response: Contact your system programmer.

System Programmer Response: If you want to use the external message catalog, correct the indicated error. If the default messages are acceptable, no action is necessary.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8212I Pagtsnmp subagent: waiting for group *mib_tree*

Explanation: The pagtsnmp subagent attempted to register the specified MIB tree *mib_tree*. The MIB tree is a set of SNMP variables that will be processed by the pagtsnmp subagent. However, another DPI subagent had already registered the requested MIB tree with a higher priority, or the pagtsnmp subagent had successfully registered the specified MIB tree, but the SNMP Agent received a later registration from another subagent which requested a higher priority than that with which the pagtsnmp subagent was currently registered.

System Action: The subagent will continue processing for the other MIB trees supported by the subagent. If the failing MIB tree becomes available at a later time the subagent will begin processing for that MIB tree.

User or Operator Response: Contact your system programmer.

System Programmer Response: If it is acceptable for a DPI subagent program other than the pagtsnmp subagent

to provide the processing for the MIB variables in the specified MIB tree, then no action is necessary. Otherwise, the other DPI subagent must be ended in order for the pagtsnmp subagent to provide the processing for the variables in the specified MIB tree.

Source Data Set: PAGTSNMP

Procedure Name: doRegister, doUnreg

EZZ8213I Pagtsnmp subagent: group *mib_tree* unregistered by SNMP Manager

Explanation: The pagtsnmp subagent received a notification that the MIB tree *mib_tree* was unregistered. This action was initiated by a request from an SNMP Manager.

System Action: The subagent will continue processing for the other MIB trees supported by the subagent. The variables under the MIB tree unregistered by the SNMP Manager will no longer be available.

User or Operator Response: Contact your system programmer.

System Programmer Response: If it is acceptable that the specified MIB tree is no longer available, then no action is necessary. To regain the unregistered MIB tree, the pagtsnmp subagent must be closed by an SNMP Manager. This can be accomplished by setting the saStatus for the pagtsnmp subagent to a value of **2**. This will cause the pagtsnmp subagent to disconnect from the SNMP Agent, then reconnect and reregister all of its supported MIB trees, including any which were previously unregistered by an SNMP Manager request.

Source Data Set: PAGTSNMP

Procedure Name: doUnreg

EZZ8214I PAGTSNMP SUBAGENT: COULD NOT ESTABLISH AFFINITY WITH *tcpip_name* - *errno errnojr*

Explanation: The pagtsnmp subagent attempted to use the UNIX System Services socket call, setibmopt(), to associate itself with the TCP/IP instance *tcpip_name*. This TCP/IP name should be the started procedure name (or identifier if the S member.identifer format of the MVS Start command was used) of the TCP/IP instance under which the pagtsnmp subagent is initializing. The setibmopt call failed with the displayed *errno* and *errnojr*.

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

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Ensure that you specified a correct TCP/IP instance name for the pagtsnmp subagent. Issue the D TCPIP command and verify that the TCP/IP name specified for the pagtsnmp subagent matches one of the names displayed. If this does not resolve the problem, contact your system programmer.

System Programmer Response: Possible reasons for this message include:

- The TCPIPJOBNAME parameter in the TCPIP DATA file doesn't match the name of a started TCP/IP stack. Check that the correct resolver TCPIP DATA file is being used, and that the name specified via the TCPIPJOBNAME parameter is correct.
- The TCP/IP instance's name was not defined correctly to UNIX System Services. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure UNIX System Services. Ensure that the TCP/IP started procedure name (or identifier if the S member.identifier format of the MVS Start command was used) matches the SUBFILESYSTYPE NAME.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8215I PAGTSNMP SUBAGENT: LOST CONNECTION TO SNMP AGENT

Explanation: The pagtsnmp subagent was connected to the SNMP Agent, but the connection was broken.

System Action: The subagent will try to reconnect to the SNMP Agent until successfully reconnected.

User or Operator Response: Contact your system programmer.

System Programmer Response: If the SNMP Agent job is not active, restart the SNMP Agent. If the SNMP Agent is currently active, the pagtsnmp subagent should automatically reconnect to the agent. If it does not, stop the SNMP Agent and restart it.

Source Data Set: PAGTSNMP

Procedure Name: doOpenAndRegister

EZZ8216I PAGTSNMP SUBAGENT: CONNECTED TO SNMP AGENT

Explanation: The pagtsnmp subagent established a connection with the SNMP Agent.

System Action: The pagtsnmp subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: doOpenAndRegister

EZZ8217I Pagtsnmp subagent: Using loopback to connect to SNMP agent

Explanation: The pagtsnmp subagent was unable to resolve the local host address and is using the loopback address to connect to the SNMP agent instead of to the host address.

The pagtsnmp subagent uses the gethostid() socket function to retrieve the local host address. The IP address returned by this function is the primary interface address of the TCP/IP stack associated with the subagent. If the returned IP address is loopback, or the gethostid() function failed, then the subagent uses loopback to connect to the SNMP agent and issues this message.

System Action: The subagent tries to connect to the agent using the loopback address.

User or Operator Response: Contact the TCP/IP administrator.

System Programmer Response: If the pagtsnmp subagent should not use loopback to connect to the agent, ensure that there is a non-loopback IP address defined as the primary interface to the TCP/IP stack associated with the subagent. The primary interface is either the first LINK in the HOME list, or the LINK specified on a PRIMARYINTERFACE profile statement. You can use the TSO NETSTAT HOME or OS/390 UNIX onetstat/netstat -h commands to determine which LINK is the primary interface for a stack.

If the loopback address is used to connect to the agent, and a password other than the SNMP agent -c default password is used by the subagent when connecting, then the password used by the subagent must be defined for the loopback address 127.0.0.1 in the SNMP agent PW.SRC or SNMPD.CONF file.

If a non-loopback IP address is defined as the primary interface, re-create the problem with the pagtsnmp subagent -d trace option. The trace messages indicate the error that the subagent received when it attempted to retrieve the primary interface IP address.

Source Data Set: PAGTSNMP.C

Procedure Name: doGethostname

EZZ8218I PAGTSNMP SUBAGENT: UNRECOGNIZED MODIFY REQUEST

Explanation: The pagtsnmp subagent was unable to process a modify command.

System Action: None.

User or Operator Response: Correct the syntax of the command. Refer to *z/OS Communications Server: IP Configuration Reference* for a description of the valid parameters.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: mvs_command_handler()

EZZ8219I PAGTSNMP SUBAGENT: MODIFY REQUEST COMPLETED

Explanation: The pagtsnmp subagent successfully processed a modify command.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: mvs_command_handler()

EZZ8220I PAGTSNMP SUBAGENT: UNSUPPORTED COMMAND RECEIVED

Explanation: The pagtsnmp subagent received a command that was neither a MODIFY or STOP command.

System Action: None.

User or Operator Response: Contact your system programmer.

System Programmer Response: Contact the IBM support center.

Source Data Set: PAGTSNMP

Procedure Name: mvs_command_handler()

EZZ8221I Pagtsnmp subagent: community string is too long

Explanation: The pagtsnmp subagent was unable to process a -c option on the command line. A community string must be less than or equal to 32 characters in length.

System Action: None.

User or Operator Response: Restart pagtsnmp with a correct community string.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: parseParms()

EZZ8222I Pagtsnmp subagent: running as jobname *jobname*

Explanation: The pagtsnmp subagent is reporting its jobname as *jobname*.

System Action: The pagtsnmp subagent continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8223I PAGTSNMP SUBAGENT: UNABLE TO CONNECT TO SNMP AGENT

Explanation: The pagtsnmp subagent initialization was not successful. Refer to syslogd for additional messages which may further describe the particular reason.

System Action: The subagent will try to connect to the SNMP Agent periodically.

User or Operator Response: Contact your system programmer.

System Programmer Response: If the SNMP Agent job is not active, restart the SNMP Agent. If the SNMP Agent is currently active, the pagtsnmp subagent should automatically reconnect to the agent. If it does not, stop the SNMP Agent and restart it.

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8224I PAGTSNMP SUBAGENT: TRACING IS SET TO *trace_level*

Explanation: This is a response to a modify command that indicates the current *trace_level*.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: mvs_command_handler

EZZ8225I PAGTSNMP SUBAGENT: COULD NOT DETERMINE TCPIP JOBNAME, USING DEFAULT OF 'INET'

Explanation: The pagtsnmp subagent could not determine the jobname for the TCP/IP stack that it is to associate itself with. A default value of 'INET' will be used for TCPIPjobname.

System Action: The pagtsnmp subagent continues processing.

User or Operator Response: None.

System Programmer Response: In a UNIX System Services INET environment, no action is necessary. In a UNIX System Services Common INET environment, the TCPIPjobname should be set in the appropriate resolver configuration file or data set, in order for the pagtsnmp subagent to communicate with a particular stack. The search order used to locate the resolver configuration data set or file is described in the *z/OS Communications Server: IP Configuration Reference*.

A pagtsnmp subagent must be associated with a single TCP/IP instance, since it retrieves information from a single TCP/IP instance to implement its MIB objects.

Source Data Set: PAGTSNMP.C

Procedure Name: main

EZZ8226I PAGTSNMP SUBAGENT: IOCTL ERROR DETECTED, PAGTSNMP ENDING

Explanation: This message indicates that an error was received from an ioctl call which caused the subagent to abnormally terminate.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Collect debug information via the pagtsnmp -d 2 parameter and forward the results to your system programmer for resolution.

System Programmer Response: Contact the IBM support center with the collected debug information.

Source Data Set: PAGTSMIB

Procedure Name: getioctl

EZZ8227I PAGTSNMP SUBAGENT: PAGENT API ERROR *code***.**

Explanation: The pagtsnmp subagent is exiting abnormally because of a Policy Agent (Pagent) API error. The error is described by *code*.

CODE	Meaning
15	The pagtsnmp subagent is not executing in superuser state.
16	The pagtsnmp subagent could not allocate storage to execute the request.
17	Pagent could not allocate storage to execute the pagtsnmp subagent request.
18	Pagent encountered an internal error and cannot execute the pagtsnmp subagent request.
19	The pagtsnmp subagent encountered an internal error and cannot continue executing the pagtsnmp subagent request.

20 The pagtsnmp subagent is not registered to Pagent.

30 or Greater
The pagtsnmp subagent lost its connection to Pagent.

System Action: The pagtsnmp subagent stops.

User or Operator Response: Contact your system programmer.

System Programmer Response: The actions required for the following errors are:

CODE	Action
15	Either set up the pagtsnmp subagent to be executed as a superuser or re-create the problem with the pagtsnmp subagent -d trace option.
16	Increase the region size for the pagtsnmp subagent.
17	Increase the region size for Pagent or re-create the problem with the pagent -d trace option or a LogLevel 511 statement in the pagent configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.
18, 20	Re-create the problem with the pagent -d trace option or a LogLevel 511 statement in the pagent configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.
19	Re-create the problem with the pagtsnmp subagent -d trace option.
30 or Greater	Validate that Pagent is executing. Re-create the problem with the pagent -d trace option or a LogLevel 511 statement in the pagent configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

Take the necessary corrective action based on the error code. Restart Pagent or reissue the pagtsnmp subagent request if desired.

Source Data Set: PAGTSMIB

Procedure Name: refreshPolicyObject

Procedure Name: buildPolicyStatsTable

Procedure Name: refreshStatsValues

Source Data Set: PAGTSNMP

Procedure Name: main

EZZ8228I PAGTSNMP SUBAGENT: UNABLE TO CONNECT TO PAGENT

Explanation: The pagtsnmp subagent was not able to connect to the Policy Agent (Pagent).

System Action: The pagtsnmp subagent periodically attempts to connect to Pagent.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that Pagent is started correctly or re-create the problem with the pagent -d trace option or a LogLevel 511 statement in the pagent configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

Source Data Set: PAGTSNMP

Procedure Name: papiConnect

EZZ8229I PAGTSNMP SUBAGENT: CONNECTED TO PAGENT

Explanation: The pagtsnmp subagent connected to the Policy Agent (Pagent).

System Action: The pagtsnmp subagent waits for requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGTSNMP

Procedure Name: doOpenAndRegister

EZZ8252I UNABLE TO OPEN MESSAGE CATALOG ″SPXMSG.CAT″ : *error*

Explanation: The D TCPIP,,SYSPLEX command was unable to open the command message catalog ″spxmsg.cat″ in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be ″NLSPATH=/usr/lib/nls/msg/%L/%N″.

System Action: The command will use the internal default messages instead of the message from the external message catalog.

User or Operator Response: If use of the external message catalog is desired, correct the indicated error. If the default messages are acceptable, no action is necessary.

System Programmer Response: If use of the external message catalog is desired, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *z/OS C/C++ Run-Time Library Reference* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the *z/OS UNIX System Services Programming Tools* book.

If the default messages are acceptable, no action is necessary. If correcting the problem caused the message catalog in the message catalog directory to change, you will have to recycle the TCP/IP stack to activate the changes.

Source Data Set: EZACDSPX

Procedure Name: cdDsp

EZZ8253I DISPLAY SYSPLEX TERMINATED DUE TO UNEXPECTED ERROR

Explanation: An unexpected event caused the command to terminate.

System Action: The Display Sysplex command terminates.

User or Operator Response: None.

System Programmer Response: See previously displayed error messages, related to this command, for an explanation of the error. If there are no previous error messages, reissue the command specifying the DEBUG option. This will cause trace messages to be written to the TCP/IP SYSPRINT data set. Save the trace messages and contact IBM support.

Source Data Set: EZACDSPX

Procedure Name: cdDsp

EZZ8254I *ioctl* **FAILED WITH ERROR :** *error* **(***errno***/***errnojr***).**

Explanation: The Display Sysplex command issued the SIOCGIBMOPT ioctl for the *ioctl* subtype. The ioctl call failed with the specified error.

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

errnojr is the hexadecimal UNIX System Services Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The Display Sysplex command terminates.

User or Operator Response: Correct the indicated error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

System Programmer Response: Correct the indicated error. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the socket errors.

Source Data Set: EZACDSPX

Procedure Name: procVIPA

EZZ8255I DISPLAY SYSPLEX USING NAME NODENAME FOR MVS SYSTEM NAME

Explanation: The command was unable to retrieve the MVS system name using the uname() function.

System Action: The Display Sysplex command continues using a value of NODENAME for the MVS system name, but some information may be missing from the report.

User or Operator Response: None.

System Programmer Response: Reissue the command specifying the DEBUG option. This will cause trace messages to be written to the TCP/IP SYSPRINT data set. Save the trace messages and contact IBM support.

System Programmer Response: None.

Source Data Set: EZACDSPX

Procedure Name: cdDsp

EZZ8260I SYSPLEX *versionRelease*

Explanation: This message displays the current version and release for the command. The message is followed by the output for the requested command report.

System Action: The Display Sysplex command continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZACDSPX

Procedure Name: procVIPA, contVIPA

EZZ8301I VIPA *ip_addr* **TAKEN OVER FROM** *tcp_jobname* **ON** *mvsname*

Explanation: The VIPA appeared in a VIPADYNAMIC VIPABACKUP list for the stack issuing the message, and the VIPA was deleted (VIPADYNAMIC VIPADELETE or IOCTL DELETE) from the stack where it was active or the other stack terminated. The stack issuing the message was first in the backup list and now defined and activated the VIPA.

ip_addr is the IP address of the Dynamic VIPA.

tcp_jobname is the name of the job associated with the procedure that was used to start TCP/IP.

mvsname is the name of the MVS system where the TCP/IP job is.

System Action: TCP/IP continues.

EZZ8302I VIPA *ip_addr* **TAKEN FROM** *tcp_jobname* **ON** *mvsname*

Explanation: The stack issuing this message activated a Dynamic VIPA that was active on another stack. The other stack deactivated the DVIPA.

ip_addr is the IP address of the Dynamic VIPA.

tcp_jobname is the name of the job associated with the procedure that was used to start TCP/IP.

mvsname is the name of the MVS system where the TCP/IP job is.

System Action: TCP/IP continues.

EZZ8303I VIPA *ip_addr* **GIVEN TO** *tcp_jobname* **ON** *mvsname*

Explanation: The TCP/IP issuing the message had the Dynamic VIPA in active state. Another TCP/IP configured the same DVIPA and this stack deactivated the DVIPA so the other stack can activate it.

ip_addr is the IP address of the Dynamic VIPA.

tcp_jobname is the name of the job associated with the procedure that was used to start TCP/IP.

mvsname is the name of the MVS system where the TCP/IP job is.

System Action: TCP/IP continues.

EZZ8304I VIPA *ip_addr* **SURRENDERED TO** *tcp_jobname* **ON** *mvsname*

Explanation: The TCP/IP that issued this message detected that its definition for the specified Dynamic VIPA (IPaddr) conflicts with the same Dynamic VIPA defined on the specified TCP/IP (TCPJobnm), and deleted its Dynamic VIPA. If the deleted Dynamic VIPA was active, connections might have been broken. The TCP/IP to which the the Dynamic VIPA was surrendered continues to have the Dynamic VIPA defined and active. The conflict that caused the TCP/IP to delete a Dynamic VIPA might have occurred from any of the following situations:

- Two TCP/IPs activate the same Dynamic VIPA nearly simultaneously.
- One TCP/IP defines a Dynamic VIPA as VIPABACKUP at nearly the same time that another TCP/IP activates the same Dynamic VIPA using the SIOCSVIPA IOCTL or a BIND. The VIPABACKUP Dynamic VIPA is deleted.
- An SIOCSVIPA IOCTL is used to define a Dynamic VIPA that another TCP/IP had previously activated using the SIOCSVIPA IOCTL or a BIND. The Dynamic VIPA that was previously activated using the SIOCSVIPA IOCTL or a BIND is deleted on that TCP/IP.

ip_addr is the IP address of the Dynamic VIPA.

tcp_jobname is the name of the job associated with the procedure that was used to start TCP/IP.

mvsname is the name of the MVS system where the TCP/IP job is.

System Action: TCP/IP continues.

User or Operator Response: Restart the reconfigured application or change profile definitions and issue the VARY TCPIP,,OBEYFILE command to the appropriate TCP/IPs.

System Programmer Response: Resolve the Dynamic VIPA conflict among the TCP/IPs.

EZZ8305I VIPA *ip_addr* **REJECTED - ACTIVE AT** *tcp_jobname* **ON** *mvsname*

Explanation: Failure indication that a VIPA with IP address *ip_addr* in a VIPADYNAMIC VIPADEFINE or VIPADYNAMIC VIPABACKUP list could not be configured because the VIPA was already activated with IOCTL or BIND on the TCP/IP identified by TCPJOBNAME, running on the MVS image named MVSNAME. The IP address was not activated for Automated Takeover (VIPADEFINE).

ip_addr is the IP address of the Dynamic VIPA.

tcp_jobname is the name of the job associated with the TCP/IP procedure that has the VIPA active.

mvsname is the name of the MVS system where the TCP/IP job is.

System Action: TCP/IP continues.

User or Operator Response: Either remove IPaddress in a VIPADEFINE list, or delete the Dynamic VIPA from the other TCP/IP identified in this message. Then try the command again.

System Programmer Response: Determine which TCP/IP stack should really be supporting the designated Dynamic VIPA.

EZZ8306I VIPA *ip_addr* **CANNOT BE DELETED**

Explanation: The specified IPaddress is one of the following:

- not currently defined to this stack
- a physical IP address
- a VIPA defined statically (not in VIPADYNAMIC VIPADEFINE or VIPADYNAMIC VIPABACKUP lists, and not activated usingIOCTL or BIND-specific within a defined VIPARANGE)

Only Dynamic VIPAs defined using VIPADEFINE or VIPABACKUP or using IOCTL or BIND() within a defined VIPARANGE may be deleted with VIPADELETE.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues.

User or Operator Response: If the IPaddress was incorrectly specified, correct the error and try the command or activation again.

EZZ8307I *ip_addr* **IS ALREADY DEFINED**

Explanation: The IP address *ip_addr* specified in a VIPADYNAMIC VIPADEFINE or VIPADYNAMIC VIPABACKUP list is already defined in a HOME list or IPCONFIG DYNAMICXCF on this stack. The address is ignored (rejected from the VIPADEFINE list in which it was defined), but other addresses in the VIPADEFINE or VIPABACKUP list are processed.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues.

User or Operator Response: If the IPaddress was incorrectly specified, correct the error and try the command or activation again.

EZZ8308I *ip_addr* **NOT DEFINED - OWNED BY** *tcp_jobname* **ON** *mvsname*

Explanation: The IP address *ip_addr* is defined in a HOME list entry or IPCONFIG DYNAMICXCF entry in another stack designated by tcpjobname and mvsname. The VIPA is deleted from the VIPADYNAMIC VIPADEFINE or VIPADYNAMIC VIPABACKUP list in which it was defined, but other VIPAs in the VIPADEFINE or VIPABACKUP list are processed.

ip_addr is the IP address.

tcp_jobname is the name of the job associated with the TCPIP procedure that owns the IP address.

mvsname is the name of the MVS system where the TCP/IP job is.

System Action: TCP/IP continues.

User or Operator Response: Either correct this stack's profile so as not to contain IPaddress in a VIPADYNAMIC list, or delete the IP address from the designated other stack, and then try the command or activation again.

System Programmer Response: Determine which TCP/IP stack should really be supporting the designated Dynamic VIPA.

EZZ8309I **TOO MANY VIPAS -** *ip_addr* **REJECTED**

Explanation: There are already 64 active and backup dynamic/moveable VIPAs on this stack defined in a combination of VIPADYNAMIC VIPADEFINE and VIPADYNAMIC VIPABACKUP lists, and requested within defined nets in VIPADYNAMIC VIPARANGE (by SIOCSVIPA IOCTL or BIND). The IP address is deleted from the VIPADEFINE or VIPABACKUP list in which it appeared, but other VIPAs in the list are processed as normal.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues.

User or Operator Response: Correct the appropriate definitions and try the command or activation again.

System Programmer Response: Reduce the number of defined or backup Dynamic VIPAs for this stack.

EZZ8310I **VIPARANGE** *ip_addr* **REJECTED - TOO MANY VIPARANGES**

Explanation: There might be at most 64 unique VIPARANGE definitions (as defined by both Network Prefix and Address Mask) active on a stack at any time.

ip_addr is the IP address used to define a VIPARANGE.

System Action: TCP/IP continues.

User or Operator Response: Correct the appropriate definitions and try the command or activation again.

System Programmer Response: Remove existing VIPARANGE definitions, or consolidate several existing ones into larger subnets, networks, or supernets.

EZZ8311I VIPARANGE *addr_mask ip_addr* **CANNOT BE DELETED**

Explanation: The VIPARANGE specified with address mask *addr_mask* and IP address *ip_addr* is not defined on the receiving TCP/IP, and thus cannot be deleted.

addr_mask is the address mask used to define a VIPARANGE.

ip_addr is the IP address used to define a VIPARANGE.

System Action: TCP/IP continues.

User or Operator Response: Display the current list of defined VIPARANGEs, if necessary. Correct the addrmask or the ipaddr or both, and try the command or activation again.

EZZ8312I VIPA *ip_addr* **MAY NOT BE CHANGED WITH** *vipadynamic*

Explanation: The Dynamic VIPA IPaddr is already defined and active on the current stack, and cannot be changed by VIPADEFINE or VIPABACKUP directly. The *vipadynamic* field can be either VIPADEFINE or VIPABACKUP.

ip_addr is the IP address used of the Dynamic VIPA.

vipadynamic is the VIPADYNAMIC statement being processed.

System Action: TCP/IP continues.

User or Operator Response: If the IPaddr was incorrectly specified, correct the error and try the command or activation again. If the IPaddr is correct and you want to change how the Dynamic VIPA is defined, you must use a VIPADELETE before the VIPADEFINE/VIPABACKUP. CAUTION: VIPADELETE will break any connections that might exist.

EZZ8313I CANNOT ACTIVATE *ip_addr* **-** *name* **IN USE**

Explanation: An attempt was made to define and activate a Dynamic VIPA (using VIPADEFINE, takeover while defined as VIPABACKUP, VIPARANGE IOCTL, or VIPARANGE BIND) and the constructed device or link name *name* is already defined on this TCP/IP. If *name* starts with VIPD, then the attempt was to create a DEVICE. If the *name* starts with VIPL, then a LINK name was being built. In either case, activation of Dynamic VIPA IPaddr failed. NOTE: If *name* starts with VIPD, then both VIPDxxxxxx and VIPLxxxxxx (where 'xxxxxx' is the hexadecimal representation of IPaddr) must not already be defined. If the check for VIPDxxxxxxxx finds it already defined, no additional check is made for VIPLxxxxxxxx at that time.

ip_addr is the IP address of the Dynamic VIPA.

name is the generated device or link name for the Dynamic VIPA.

System Action: TCP/IP continues.

User or Operator Response: Correct the appropriate definitions and try the command or activation again.

System Programmer Response: Ensure the none of the static definitions contain a link name of VIPLxxxxxxxx or a device name of VIPDxxxxxxxx.

EZZ8314I DVIPA action for ipaddress failed with ERRNO=*errno* **ERRNO2=***errnojr*

Explanation: The MODDVIPA utility failed to create or delete a Dynamic VIPA.

action is the utility function and socket call that failed. ipaddress is the Dynamic VIPA address or TCPIP stack name that is not valid.

errno is the Sockets Return Code. These return codes are listed and described in the *z/OS Communications Server: IP and SNA Codes*.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: TCP/IP continues. The utility ends.

User or Operator Response: If the IP address or TCPIP stack name was incorrectly specified, correct the error and run the utility again, otherwise contact your system programmer.

System Programmer Response: See the *z/OS UNIX System Services Messages and Codes* for more information about the *errnojr*. See the *z/OS Communications Server: IP and SNA Codes* for the specific reason for failure reported by *errno*.

Source Data Set: EZBXFDVP

Procedure Name: main

EZZ8320I Usage: host [-d] [-h|-?] *host*

Explanation: This message shows the syntax for the host command. It is displayed when incorrect parameters were specified or when the -h or -? options were specified.

host This is the DNS host name or numeric address string for the host to query.

-d This option specifies that debugging messages should be displayed. This may be useful for problem diagnosis.

-h or -?
These options request that this message be displayed.

System Action: The command terminates.

User or Operator Response: Run the command again with the correct parameters.

System Programmer Response: None.

EZZ8321I *hostname* **has addresses** *address-list*

Explanation: *address-list* is a list of IP addresses associated with the specified host or **(none)** if no IP addresses were found but the host name was valid.

System Action: The command continues.

User or Operator Response: None.

System Programmer Response: None.

EZZ8322I aliases: *alias-list*

Explanation: *alias-list* is a list of aliases for the specified host. This message will be displayed only if one or more aliases were found.

System Action: The command continues.

User or Operator Response: None.

System Programmer Response: None.

EZZ8330I Usage: hostname [-s] [-c|-g|-r] [-p *stackname***] [-d] [-h|-?]**

Explanation: This message shows the syntax for the hostname, domainname, and dnsdomainname commands. It is displayed when incorrect parameters were specified or when the -h or -? options were specified.

-s This option, only valid for the hostname command, specifies that hostname should omit the DNS domain name and print only the host-specific portion.

-c This option, the default, specifies that the host name should be retrieved from the resolver configuration file.

-g This option specifies that the host name should be retrieved using the gethostname() system call.

-r This option specifies that the host name should be retrieved from the DNS server, with the query based on the results of the gethostname() system call.

-p *stackname*
Use the specified AF_INET stack.

-d This option specifies that debugging messages should be displayed. This may be useful for problem diagnosis.

-h or -?
These options request that this message be displayed.

System Action: The command terminates.

User or Operator Response: Run the command again with the correct parameters.

System Programmer Response: None.

EZZ8331I The DNS domain name could not be determined.

Explanation: The DNS domain name could not be determined using the specified mechanism. This message is printed by the domainname and dnsdomainname commands when the DNS domain name could not be found.

If the -c option was specified (the default mechanism), this could be due to the DOMAINORIGIN keyword not being specified in the client configuration file (TCPIP.DATA).

If the -g option was specified, this could be due to gethostname() returning only the host part of the fully-qualified DNS name.

System Action: The command terminates.

User or Operator Response: Run the command again using a different mechanism (-c, -d, or -g option) for retrieving this information.

System Programmer Response: Ensure that the DOMAINORIGIN keyword is specified in the client configuration file (TCPIP.DATA).

EZZ8340I This function requires the TCP/IP base feature of OS/390.

Explanation: The function or command which issued this message is disabled because the TCP/IP base feature of OS/390 is not enabled.

System Action: The command terminates.

User or Operator Response: Contact the system programmer.

System Programmer Response: The IFAPRDxx parmlib member specifies whether or not the TCP/IP base feature is enabled. Correct the IFAPRDxx parmlib member if it is in error.

If TCP/IP_BASE is not part of your system there is no action to take and this function or command cannot be used.

EZZ8341I Error *return-code*/*reason-code* **was returned from** *libcall*: *explanation*

Explanation: The specified library or system call (*libcall*) failed. *explanation* describes the nature of the error. *return-code* is the return code (errno) in hexadecimal format from the library or system call. *reason-code* is the reason code in hexadecimal format.

System Action: Processing continues.

User or Operator Response: See *z/OS UNIX System Services Messages and Codes* for more information about *return-code* and *reason-code*.

System Programmer Response: None.

EZZ8342I *operation*: *reason*

Explanation: The specified DNS name resolution operation failed for the specified reason.

System Action: Processing continues.

User or Operator Response: Ensure that the host name or IP address is correct. Contact the system programmer if the parameters are correct but DNS name resolution fails.

System Programmer Response: Ensure that the DNS server and host name and address tables are configured correctly.

EZZ8343I This command is not supported in this environment.

Explanation: A command was used in an unsupported environment. For example, a command that is supported only in the z/OS UNIX shell was used in the TSO environment.

System Action: Processing continues.

User or Operator Response: Use this command in a supported environment. For example, if the command is supported only in the z/OS UNIX shell, switch to that environment and use the command from that environment.

System Programmer Response: None

EZZ8344I This command must be APF-authorized

Explanation: A command was used that requires APF authorization to function correctly, but the command was not running APF-authorized.

System Action: Processing continues.

User or Operator Response: Contact the system programmer to ensure that the program was installed correctly.

System Programmer Response: Ensure that the program was installed correctly. If the program executable resides in the HFS, ensure that the APF-authorized attribute is on. If the program resides in a load module, ensure that the library which contains it is APF authorized.

EZZ8345I Message catalog *filename* **could not be opened - Default messages will be used**

Explanation: The message catalog file could not be opened. See message EZZ8341I for the error codes. The command will use the default message strings instead of the message strings in the message catalog file.

filename is the name of the message catalog.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Ensure that the message catalog was installed properly.

EZZ8392I Unable to open message catalog *trapfwd.cat* **-** *additional_error_text*

Explanation: The Trap Forwarder daemon was unable to open the message catalog *trapfwd.cat* in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be NLSPATH=/usr/lib/nls/msg/%L/%N.

System Action: The Trap Forwarder daemon will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If you want to use the external message catalog, contact the system programmer. If the default messages are acceptable, no action is necessary.

System Programmer Response: If you want to use the external message catalog, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *z/OS C/C++ Run-Time Library Reference* for more information about the catopen() function call. Information regarding the NLSPATH environment variable can be found in the *z/OS UNIX System Services Programming Tools*. If the default messages are acceptable, no action is necessary.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitLogging

EZZ8393I Using catalog *catalog_file* **for TRAPFWD messages**

Explanation: The Trap Forwarder daemon located its message catalog file.

catalog_file is the name of the catalog file.

System Action: The Trap Forwarder daemon continues to initialize.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitLogging

EZZ8394I No value specified for the *startup* parameter

Explanation: When the Trap Forwarder daemon was started, a parameter was specified without a value.

startup is a command line parameter.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: Correct the error and restart the Trap Forwarder daemon with the correct parameter value.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwParseCmdLineArgs

EZZ8395I *parameter* value is out of range

Explanation: While processing the start options for the Trap Forwarder daemon, a parameter was encountered that required a numeric value within a specified range. The value specified was outside the allowable range of values for the parameter being processed.

parameter is a command line parameter.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: Correct the value specified for the parameter in error and restart the Trap Forwarder daemon.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwParseCmdLineArgs

EZZ8396I *parameter* value is not numeric

Explanation: While processing the start options for the Trap Forwarder daemon, a parameter was encountered that required a numeric value. The value specified was not numeric.

parameter is a command line parameter.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: Correct the value specified for the parameter in error and restart the Trap Forwarder daemon.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwParseCmdLineArgs

EZZ8397I Unable to open *configuration_file* - *error_text*

Explanation: The Trap Forwarder daemon was unable to open the *configuration_file*.

configuration_file is the name of the configuration file.

error_text is the error message string, containing a more specific reason for the failure.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: Correct the error and restart the Trap Forwarder daemon.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8398I Line number *linenumber* **in the** *configuration_file* **file has an IP address** *address* **which is not in the correct format - entry ignored**

Explanation: The Trap Forwarder daemon was unable to interpret the IP address correctly.

linenumber is the number of the line in the configuration file.

configuration_file is the name of the configuration file.

address is the IP address.

System Action: The Trap Forwarder daemon ignores the current statement and continues reading the next statement in the configuration file.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error and restart the Trap Forwarder daemon or issue a refresh command to refresh the configuration. For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8399I gethostbyname function failed for *address* **- entry ignored**

Explanation: The gethostbyname function failed for the indicated IP address.

address is the IP address.

System Action: The Trap Forwarder daemon ignores the current statement and continues reading the configuration file.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error and restart the Trap Forwarder daemon or issue a refresh command to refresh the configuration. For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Guide*.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8400I Line number *linenumber* **in the** *configuration_file* **file has an incorrect port number** *port_number* **- entry ignored**

Explanation: The port number at the specified line number is not valid.

linenumber is the number of the line in the configuration file.

configuration_file is the name of the configuration file.

port_number is the port number specified in the configuration file.

System Action: The line is ignored and the Trap Forwarder daemon continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the IP address in the configuration file and restart the Trap Forwarder daemon or issue a refresh command to refresh the configuration. For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8401I Line number *linenumber* **in the** *configuration_file* **file has an incorrect option** *option* **- entry ignored**

Explanation: The option at the specified line number is not valid.

linenumber is the number of the line in the configuration file.

configuration_file is the name of the configuration file.

option is the option specified in the configuration file.

System Action: The line is ignored and the Trap Forwarder daemon continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the option in the configuration file and restart the Trap Forwarder daemon or issue a refresh command to refresh the configuration. For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8402I Line number *linenumber* **in the** *configuration_file* **file contains an incorrect number of keywords - entry ignored**

Explanation: The indicated line contains an incorrect number of keywords.

linenumber is the number of the line in the configuration file.

configuration_file is the name of the configuration file.

System Action: The line is ignored and the Trap Forwarder daemon continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the entry in the configuration file and restart the Trap Forwarder daemon or issue a refresh command to refresh the configuration. For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8403I Maximum number of destinations exceeded on line *linenumber* **in the** *configuration_file* **file - entry ignored**

Explanation: The maximum number of destinations to which trap datagrams can be forwarded was exceeded.

linenumber is the number of the line in the configuration file.

configuration_file is the name of the configuration file.

System Action: The line is ignored and the Trap Forwarder daemon continues processing.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the configuration file. The maximum number could have been reached for either of the following reasons:

- The destinations where the trap originating information has to be appended.
- The destinations where the trap datagrams are forwarded without the originating information.

For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TFWINIT.C

Procedure Name: tfwReadConfigFile

EZZ8404I TRAPFWD: COULD NOT DETERMINE TCPIPJOBNAME - DEFAULTING TO INET

Explanation: The Trap Forwarder daemon could not determine the jobname for the TCP/IP stack. The default value of 'INET' is used.

System Action: Trap Forwarder daemon continues initializing.

User or Operator Response: None.

System Programmer Response: In an INET environment, no action is necessary. In a CINET environment, for the Trap Forwarder daemon to communicate with a particular stack, the TCPIPJobname should be set in the appropriate resolver configuration file or data set. Refer to the *z/OS Communications Server: IP Configuration Guide* for the search order to locate the resolver configuration data set or file. The Trap Forwarder daemon must be associated with a single TCPIP instance.

Source Data Set: TFWINIT.C

Procedure Name: tfwEstablishAffinity

EZZ8405I TRAPFWD: COULD NOT ESTABLISH AFFINITY WITH '*jobname*' (*error_code*/*reason*)

Explanation: The Trap Forwarder daemon cannot communicate with the TCP/IP stack *jobname*. The Trap Forwarder daemon attempted to use the socket call, setibmopt(), to associate itself with the TCP/IP instance *jobname*. This TCP/IP name should be the started procedure name (or identifier if the 'S member.identifer' format of the MVS Start command was used) of the TCP/IP instance with which the Trap Forwarder daemon is to be associated. The setibmopt call failed with the displayed *error_code* and *reason*.

jobname is the jobname of the Trap Forwarder daemon.

error_code is the error number.

reason is the reason code.

System Action: The Trap Forwarder daemon ends abnormally.

User or Operator Response: Most likely, the TCP/IP instance's name was not defined correctly to OMVS. Check the SUBFILESYSTYPE NAME for the corresponding TCP/IP instance in the BPXPRMxx member that was used to configure OMVS. Ensure that the TCP/IP started procedure name (or identifier if the 'S member.identifier' format of the MVS Start command was used) matches the SUBFILESYSTYPE NAME. Recycle OMVS or TCP/IP if a change is necessary. If none of the above error conditions exist contact your system programmer.

System Programmer Response: For the Trap Forwarder daemon to communicate with a particular stack, the *jobname* (as determined by the system variable TCPIPjobname) must match ″xxxxx″ where ″xxxxx″ is set in the BPXPRMPx member used to start OMVS. ″xxxxx″ is set in the SUBFILESYSTYPE NAME(xxxxx) for ENTRYPOINT(EZBPFINI). In order to establish an affinity with a corresponding TCP/IP stack, the Trap Forwarder daemon uses the setibmopt call. Correct the error indicated by *error_code* and *reason*. For more information about error_code and reason, see *z/OS UNIX System Services Messages and Codes*.

Source Data Set: TFWINIT.C

Procedure Name: tfwEstablishAfiinity

EZZ8406I Sigaction for *signal* failed - *error_text (error_code/reason)*

Explanation: The Trap Forwarder daemon issued a sigaction function that failed. The *error_text* will provide more information about the cause of the error.

signal is the name of the signal.

error_text is the error message string.

System Action: Trap Forwarder daemon continues initializing.

User or Operator Response: None.

System Programmer Response: Correct the problem indicated by *error_text*. Refer to the *z/OS C/C++ Run-Time Library Reference* for further explanation of the sigaction function errors.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitSignalHandlers

EZZ8407I Unable to allocate memory

Explanation: The Trap Forwarder daemon attempted to allocate memory to receive trap datagrams. Memory could not be obtained.

System Action: The Trap Forwarder daemon initialization terminates.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why memory was not available. Correct the problem and restart the Trap Forwarder daemon.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitialize

EZZ8408I TRAPFWD: INITIALIZATION TERMINATED - TCP/IP BASE FEATURE OF OS/390 IS NOT ENABLED

Explanation: The Trap Forwarder daemon issued a query for TCP/IP but the IFAPRDxx parmlib member does not indicate that the feature TCP/IP_BASE is enabled. This Trap Forwarder daemon will only run if TCP/IP_BASE is enabled.

System Action: The Trap Forwarder daemon initialization terminates.

User or Operator Response: Contact the system programmer.

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

Source Data Set: TFWINIT.C

Procedure Name: tfwCheckStackEnabled

EZZ8409I TRAPFWD: INITIALIZATION COMPLETE

Explanation: The Trap Forwarder daemon completed initialization and is ready to receive and forward trap datagrams.

System Action: The Trap Forwarder daemon is functioning.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitialize

EZZ8410I Maximum size of trap datagram exceeded - trap datagram discarded

Explanation: The Trap Forwarder daemon received a trap datagram that has a size greater than the size specified during startup. The trap datagram is discarded.

System Action: The Trap Forwarder daemon continues to listen for trap datagrams.

User or Operator Response: Contact the system programmer.

System Programmer Response: If a trap datagram of a larger size needs to be forwarded, use the -l startup option to configure the Trap Forwarder daemon. For information about configuring the Trap Forwarder daemon, refer to the *z/OS Communications Server: IP Configuration Guide*.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitialize

EZZ8411I *function* **function failed - error code:** *error_code* **reason:** *reason*

Explanation: The system function failed. *error_code* and *reason* will provide more information about the cause of the error.

function is the system function that failed.

error_code is the error number.

reason is the reason code.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why the system function failed. For more information about error_code and reason, see *z/OS C/C++ Run-Time Library Reference*.

Source Data Set: TFWINIT.C, TFWMAIN.C

Procedure Name: tfwProcessIncomingTrapPkt, tfwCreateSockets

EZZ8412I TRAPFWD: TRACING IS SET TO *setting*

Explanation: This is a response to a modify command that indicates the current trace setting.

setting is the debug level.

System Action: The Trap Forwarder daemon continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWMAIN.C

Procedure Name: tfwProcessMVSCommand

EZZ8413I TRAPFWD: UNRECOGNIZED MODIFY REQUEST

Explanation: This is a response to a modify command that could not be serviced because the request was not recognized.

System Action: The Trap Forwarder daemon continues.

User or Operator Response: Correct the error and issue the modify command again.

System Programmer Response: None.

Source Data Set: TFWMAIN.C

Procedure Name: tfwProcessMVSCommand

EZZ8414I TRAPFWD: MODIFY REQUEST COMPLETED

Explanation: This is a response to a modify command that completed.

System Action: The Trap Forwarder daemon continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWMAIN.C

Procedure Name: tfwProcessMVSCommand

EZZ8416I TRAPFWD: CONFIGURATION REFRESH FAILED

Explanation: The Trap Forwarder daemon tried to refresh the configuration by reading the configuration file again but failed.

System Action: The Trap Forwarder daemon continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the configuration file exists and is readable.

Source Data Set: TFWMAIN.C

Procedure Name: tfwProcessMVSCommand

EZZ8417I *signal* **RECEIVED - TRAPFWD DAEMON IS SHUTTING DOWN**

Explanation: The Trap Forwarder daemon received a signal and is shutting down.

signal is the name of the signal.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: sigPipeHandler, sigAbndHandler, sigTermHandler

EZZ8418I Usage: *command***<-d debug level><-p port number> <-l max packet len>**

Explanation: Shows allowed syntax for calling trapfwd.

command is the command name.

System Action: The trapfwd daemon terminates after displaying usage information.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwDisplayUsage

EZZ8419I Unsupported parameter: *parameter*

Explanation: An unsupported parameter was specified while starting the Trap Forwarder daemon.

parameter is a command line parameter.

System Action: The Trap Forwarder daemon terminates.

User or Operator Response: Correct the error and restart the Trap Forwarder daemon.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwParseCmdLineArgs

EZZ8420I The Trap Forwarder daemon is running as *jobname*

Explanation: The Trap Forwarder daemon is reporting its jobname.

jobname
is the jobname of the Trap Forwarder daemon

System Action: The Trap Forwarder daemon continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TFWINIT.C

Procedure Name: tfwInitialize

EZZ8431I PAGENT STARTING

Explanation: The Policy Agent (PAGENT) application is starting.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: main

EZZ8432I PAGENT INITIALIZATION COMPLETE

Explanation: The Policy Agent (PAGENT) completed initialization and is ready to start processing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: main

EZZ8433I PAGENT SHUTDOWN COMPLETE

Explanation: The Policy Agent (PAGENT) application is exiting normally.

System Action: PAGENT ends.

User or Operator Response: None.

System Programmer Response: Restart PAGENT if desired.

Source Data Set: PAGENT

Procedure Name: main

Source Data Set: PAGENT

Procedure Name: terminator

EZZ8434I PAGENT EXITING ABNORMALLY

Explanation: The Policy Agent (PAGENT) is exiting abnormally.

System Action: PAGENT ends.

User or Operator Response: Re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

System Programmer Response: Take the necessary corrective action based on log information indicating the cause of the abnormal exit. Restart PAGENT if desired.

Source Data Set: PAGENT

Procedure Name: main

EZZ8435I pasearch Command: Environment Error *code*

Explanation: The pasearch command is exiting abnormally due to an environment error. The error is described by *code*.

CODE	Meaning
15	pasearch is not executing in superuser state.
16	Could not allocate storage to execute pasearch command.
17	Policy Agent (PAGENT) could not allocate storage to execute pasearch command.
18	Policy Agent (PAGENT) encountered an internal error and cannot execute the pasearch request.
19	pasearch encountered an internal error and cannot continue executing the pasearch request.
20	pasearch is not registered to Policy Agent (PAGENT).
21	pasearch is not executing with the correct security level.
22	pasearch is executing with incorrect level of PAPI.DLL.

System Action: The pasearch command ends.

User or Operator Response: The actions required for the following errors are:

CODE	Action
15	Either set up pasearch to be executed as a superuser or with the correct security level, or re-create the problem with the pasearch -d trace option.
16	Increase the region size for pasearch or limit the search information returned from pasearch.
17	Increase the region size for PAGENT or re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.
18	Re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.
19	Re-create the problem with the pasearch -d trace option.
20	Re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.
21	Either set up the user of pasearch to be executed with correct security level or re-create the problem with paearch -d trace option.
22	The Policy Agent API libraries (papi.dll) must be accessible to pasearch. The LIBPATH environment variable can be set to indicate where papi.dll is found (/usr/lpp/tcpip/lib).

System Programmer Response: Take the necessary corrective action based on the error code. Restart PAGENT or re-issue pasearch if desired.

Source Data Set: PASEARCH

Procedure Name: paIssueMessage

EZZ8436I pasearch Command: Connection Error *code*

Explanation: The pasearch command encountered a connection error with the Policy Agent (PAGENT). The error is described by *code*.

CODE	Meaning
30	Connect to PAGENT failed.
31	Read from PAGENT failed.
32	Cannot read data from PAGENT.
33	Write to PAGENT failed.

34 Read from PAGENT timed out.

35 Could not open socket to PAGENT.

36 fcntl call failed to PAGENT.

System Action: The pasearch command ends.

User or Operator Response: Validate that Policy Agent (PAGENT) is executing. Re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file and pasearch -d trace option. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

System Programmer Response: Take the necessary corrective action based on log information indicating the cause of the abnormal exit. Start PAGENT if desired and re-issue pasearch.

Source Data Set: PASEARCH

Procedure Name: paIssueMessage

EZZ8437I pasearch Command: Parameter Error *code*

Explanation: The pasearch command encountered a parameter error. The error is described by *code*.

CODE	Meaning
50	A required input parameter was not passed.
51	The TcpImage was not configured to Policy Agent (PAGENT).
52	The TcpImage was greater than eight characters.
53	The PolicyFilterName or PolicyScopeName was greater than 47 characters.

System Action: The pasearch command ends.

User or Operator Response: Correct the input parameter based on the error code or re-create the problem with the pasearch -d trace option and with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

System Programmer Response: Take the necessary corrective action based on log information indicating the cause of the abnormal exit. Re-issue pasearch if desired.

Source Data Set: PASEARCH

Procedure Name: paIssueMessage

EZZ8438I PAGENT POLICY DEFINITIONS CONTAIN ERRORS

Explanation: The policies defined in a configuration file or on an LDAP server contain errors.

System Action: The policy entries in error are not applied.

User or Operator Response: Re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

System Programmer Response: Correct the PAGENT policy definition errors identified in the log and restart PAGENT with the corrected policy definitions.

Source Data Set: LDAPCLNT

Procedure Name: ReadLdapRules

Source Data Set: PINIT

Procedure Name: pinit_fetch_policy_profile

EZZ8439I PAGENT READFROMDIRECTORY STATEMENT CONTAINS ERRORS

Explanation: The Policy Agent (PAGENT) ReadFromDirectory statement in the configuration file contains errors.

System Action: The ReadFromDirectory statement is not applied and PAGENT does not attempt to connect to the LDAP server.

User or Operator Response: Re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

System Programmer Response: Correct the PAGENT configuration file errors identified in the log and restart PAGENT with the corrected configuration file.

Source Data Set: PINIT

Procedure Name: processing_Stmt_UseLDAPRules

EZZ8440I PAGENT CANNOT CONNECT TO LDAP SERVER FOR *TcpImage*

Explanation: The Policy Agent (PAGENT) could not connect to the LDAP server for *TcpImage*. This might indicate a problem with the LDAP server, or might indicate that the associated TCP/IP stack was recycled.

If the LDAP server had successfully connected, then the original policies will not be deleted from the TCP/IP stack. If no policies are changed when the LDAP server reconnects, then no new message will be displayed.

System Action: PAGENT retries connecting to the LDAP server, using a sliding retry interval that starts at 1 minute and increases up to 30 minutes at 5 minute intervals. As long as the connection attempt is unsuccessful, PAGENT cannot read LDAP policies.

User or Operator Response: If the problem persists, verify the LDAP server is running and that the correct connection parameters have been specified on the ReadFromDirectory configuration statement. If the connect attempt still fails, re-create the problem with the PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. If you are using SYSLOGD, ensure that the Syslog Daemon (syslogd) is running.

System Programmer Response: Correct the PAGENT configuration file or the LDAP server problem and restart either PAGENT or the LDAP server.

Source Data Set: LDAPCLNT

Procedure Name: ReadLdapRules

EZZ8441I PAGENT MODIFY COMMAND UNSUCCESSFUL - SYNTAX ERROR

Explanation: The Policy Agent (PAGENT) application is unable to process a MODIFY command. An incorrect parameter was specified on the command.

System Action: The Policy Agent (PAGENT) MODIFY command is ignored.

User or Operator Response: Verify the syntax of the MODIFY command and reissue the command. See the *z/OS Communications Server: IP Configuration Reference* for the syntax of the Policy Agent (PAGENT) MODIFY command.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: PAGENT_mvs_command_handler

EZZ8442I PAGENT MODIFY COMMAND UNSUCCESSFUL - INCORRECT VALUE

Explanation: The Policy Agent (PAGENT) application is unable to process a MODIFY command. An incorrect value was specified on the command.

System Action: The Policy Agent (PAGENT) MODIFY command is ignored.

User or Operator Response: Verify the correct parameter values for the MODIFY command and reissue the command. See the *z/OS Communications Server: IP Configuration Reference* for the syntax of the Policy Agent (PAGENT) MODIFY command.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: PAGENT_mvs_command_handler

EZZ8443I PAGENT MODIFY COMMAND ACCEPTED

Explanation: A MODIFY command was accepted by the Policy Agent (PAGENT).

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: PAGENT_mvs_command_handler

EZZ8444I PAGENT LOG LEVEL *loglevel* **DEBUG LEVEL** *debuglevel* **TRACE LEVEL** *tracelevel*

Explanation: This is the response to a Policy Agent (PAGENT) MODIFY procname,QUERY command.

loglevel is the value specified with the LogLevel configuration statement or with the MODIFY LOGLEVEL,LEVEL command.

debuglevel is the debug level specified with the -d startup option, or with the MODIFY DEBUG,LEVEL command.

tracelevel is the trace level specified with the -t startup option, or with the MODIFY TRACE,LEVEL command.

System Action: Processing continues.

User or Operator Response: See the *z/OS Communications Server: IP Configuration Reference* for the description of each of the displayed level values.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: PAGENT_mvs_command_handler

EZZ8445I PAGENT SHUTDOWN IN PROGRESS

Explanation: The Policy Agent (PAGENT) application is about to exit normally.

System Action: PAGENT begins to shut down.

User or Operator Response: None

System Programmer Response: None

Source Data Set: PAGENT

Procedure Name: terminator

EZZ8446I TCP CONNECTION FROM PAGENT ON *tcp_procname* **TO PAGENT ON** *remote_ip_address* **:** *remote_port* **IS NO LONGER ACTIVE**

Explanation: The Policy Agent (PAGENT) running on the distributing stack lost its TCP connection with the PAGENT running on the target stack.

tcp_procname is the tcpip jobname of the distributing stack.

remote_ip_address is the ip address of the Policy Agent running on the target stack.

remote_port is the port number used by the Policy Agent running on the target stack.

System Action: The PAGENT will not be able to obtain the QoS fractions using service level granularity from the target stack. If the target stack is removed from the distributing stack target list, no action will be taken. Otherwise, the PAGENT running on the distributing stack will try to establish a connection with the PAGENT running on the target stack.

User or Operator Response: None.

System Programmer Response: If the target stack was supposed to be removed from the distributor stack target list, then no further action is necessary. If the target stack was removed from the distributor stack list in error, then look at the system log to determine the problem. If the target stack is still in the distributor stack list, then take the

necessary corrective action, based on the information in the PAGENT log, indicating the cause of the TCP connection failure.

Source Data Set: PQOSCOLL

Procedure Name: pqos_refresh_target_cache, pqos_cleanup_target_cache, pqos_get_info_from_listeners

---

EZZ8447I THROUGHPUT QOS MAXIMUM REACHED FOR SOME SERVICE LEVELS

Explanation: The Throughput QoS maximum was reached for some service levels. This message will be issued again only if the QoS level goes back to normal and then reaches the maximum again.

System Action: None.

User or Operator Response: The log for PAGENT will indicate which service levels have reached the maximum if the debug level includes debug level 8.

System Programmer Response: Take the necessary corrective action to ensure that the expected connection and traffic distribution is occurring.

Source Data Set: PQOSCACH

Procedure Name: pqos_refresh_perf_cache

EZZ8448I PAGENT DOES NOT HAVE QOSLISTENER AND QOSCOLLECTOR PORTS DEFINED

Explanation: Policy Agent (PAGENT) needs the PAGENTQosListener and PAGENTQosCollector ports defined to collect QoS statistics with service level granularity. The /etc/services file must contain the definitions for the PAGENTQosListener and PAGENTQosCollector ports.

System Action: Processing continues, but QoS statistics with service level granularity will not be used.

User or Operator Response: If QoS statistics with service level granularity is not required, no action needs to be taken. If QoS statistics with service level granularity is required, contact your system programmer.

System Programmer Response: Modify /etc/services in each node in the sysplex to have a common PAGENTQosListener port and PAGENTQosCollector port. See the *z/OS Communications Server: IP Configuration Guide* for more information.

Source Data Set: PQOSMON

Procedure Name: policy_perf_monitor

EZZ8449I PAGENT RUNNING ON *tcp_procname* **CONNECTED TO PAGENT RUNNING ON** *remote_ip_address*:*remote_port*

Explanation: The Policy Agent (PAGENT) running on a distributing stack established a connection with the PAGENT running on the target stack to collect QoS statistics with service level granularity.

tcp_procname is the tcpip jobname of the distributing stack.

remote_ip_address is the ip address of the Policy Agent running on the target stack.

remote_port is the port number used by the Policy Agent running on the target stack.

System Action: None

User or Operator Response: None

System Programmer Response: None

Source Data Set: PQOSCOLL

Procedure Name: pqos_connect_to_listener

EZZ8450I PAGENT RUNNING ON *tcp_procname* **DECLINED TO ACCEPT A CONNECTION FROM PAGENT RUNNING ON** *remote_ip_address*:*remote_port*

Explanation: The Policy Agent (PAGENT) running on the target stack, *tcp_procname*, rejected a connection request because the request was not received from a recognized distributing stack.

tcp_procname is the tcpip jobname of the target stack.

remote_ip_address is the ip address of the Policy Agent running on the distributing stack.

remote_port is the port number used by the Policy Agent running on the distributing stack.

System Action: Processing continues

User or Operator Response: Contact the system programmer.

System Programmer Response: If the remote IP address/port is not part of the sysplex, then this might be a security breach. If it is part of the sysplex, then verify that the migration tasks for the load distribution function have all been completed. If so, then re-create problem with PAGENT -d trace option or a LogLevel 511 statement in the PAGENT configuration file. Ensure that the Syslog Daemon (syslogd) is running.

Source Data Set: PQOSLISN

Procedure Name: qosListener

EZZ8451I PAGENT RUNNING ON *tcp_procname* **CANNOT ESTABLISH CONNECTION WITH PAGENT RUNNING ON** *remote_ip_address*:*remote_port*

Explanation: The Policy Agent (PAGENT) running on the distributing stack was not able to establish a TCP connection to PAGENT running on a target stack to collect QoS Service Level statistics. Although the Policy Agent on the distributing stack will continue to retry connection establishment periodically, this message will not be issued again for this target.

tcp_procname is the tcpip jobname of the distributing stack.

remote_ip_address is the ip address of the Policy Agent running on the target stack.

remote_port is the port number used by the Policy Agent running on the target stack.

System Action: Policy Agent running on the distributing stack will not be able to collect QoS statistics using service level granularity from the target node. Aggregate QoS statistics will be used for distribution of work to this target.

User or Operator Response: If QoS statistics with service level granularity is not required from this target, no action needs to be taken. If QoS statistics with service level granularity is required, contact your system programmer.

System Programmer Response: The PAGENT log indicates the reason for the connection failure. Take necessary corrective action based on the information indicating the cause of the TCP connection failure.

Source Data Set: PQOSCOLL.C

Procedure Name: pqos_connect_to_listener

EZZ8453I TCPIP STORAGE

Explanation: TCP/IP issues this message as part of a group of messages in response to a DISPLAY TCPIP,,STOR command. This is the first message in the group. A complete description of the message group follows:

```
EZZ8453I TCPIP STORAGE
EZZ8454I jobname STORAGE     CURRENT MAXIMUM  LIMIT
EZZ8455I jobname storagetype current maximum  limit
EZZ8459I DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY
```

EZZ8453I

This message identifies the type of information shown in the message group.

EZZ8454I

This message is a header message for EZZ8455I.

jobname is the jobname associated with the procedure used to start TCP/IP.

EZZ8455I

This message contains storage totals.

jobname is the jobname associated with the procedure used to start TCP/IP.

storagetype is the storage type is ECSA for TCP/IP's use of extended common storage area, or POOL for TCP/IP's use of pooled private storage in the TCP/IP address space.

current is the amount of storage currently allocated by TCP/IP. The value will end with either K to indicate 1024 bytes or M to indicate 1048576 bytes.

maximum is the maximum amount of storage TCP/IP ever allocated since it was started. The value will end with either K to indicate 1024 bytes or M to indicate 1048576 bytes.

limit is the storage limit that TCP/IP will allow. This limit is defined on the GLOBALCONFIG profile statement. See the *z/OS Communications Server: IP Configuration Reference* for more information. The value will end with either K to indicate 1024 bytes or M to indicate 1048576 bytes, or the value will be NOLIMIT if the storage does not have a limit.

EZZ8459I

This message is displayed when the D TCPIP,procname,STOR,MODULE=modid command completed.

System Action: TCP/IP continues.

User or Operator Response: None

System Programmer Response: None

Source Data Set: TCPIP

Procedure Name: EZACDDSU

EZZ8454I *jobname* **STORAGE CURRENT MAXIMUM LIMIT**

Explanation: This message is issued as part of a message group. See message EZZ8453I for a complete description of the message group.

EZZ8455I *jobname storagetype current maximum limit*

Explanation: This message is issued as part of a message group. See message EZZ8453I for a complete description of the message group.

EZZ8456I TCPIP MODULE STORAGE

Explanation: This message is displayed in response to the D TCPIP,procname,STOR,MODULE=modid command.

System Action: TCP/IP continues.

User or Operator Response: None.

Message Destination: None.

System Programmer Response: None.

+ *offset modprg1 modprg2 modprg3 modprg4 *tranlate*

Explanation: This message is displayed in response to the D TCPIP,procname,STOR,MODULE=modid command.

offset is the current display offset.

modprg1 indicates four bytes of the specified module.

modprg2 indicates four bytes of the specified module.

modprg3 indicates four bytes of the specified module.

modprg4 indicates four bytes of the specified module.

tranlate is the module line translated to EBCDIC.

System Action: TCP/IP continues.

User or Operator Response: None.

Message Destination: None.

System Programmer Response: None.

EZZ8459I DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY

Explanation: This message is displayed when the D TCPIP,procname,STOR,MODULE=modid command completed.

System Action: TCP/IP continues.

User or Operator Response: None.

Message Destination: None.

System Programmer Response: None.

EZZ8460I *modid* **AND ITS STORAGE CANNOT BE DISPLAYED**

Explanation: This message is displayed in response to the D TCPIP,procname,STOR,MODULE=modid command. when an undisplayable module was specified.

modid is the name of the specified module.

System Action: TCP/IP continues.

User or Operator Response: Re-enter the D TCPIP,procname,STOR,MODULE=modid command with a displayable module name.

Message Destination: None.

System Programmer Response: None.

EZZ8461I *ip_addr* **IS NOT A ROUTABLE DYNAMIC VIPA**

Explanation: This message is additional information for message EZZ8469I. A VIPADISTRIBUTE DEFINE statement is in a profile or VARY OBEY file, but the specified IP address is not configured on this stack as a VIPADEFINE or VIPABACKUP dynamic VIPA.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is rejected.

System Programmer Response: If you intend to distribute the specified IP address, ensure that it was configured with a VIPADEFINE or VIPABACKUP statement before the VIPADISTRIBUTE statement is processed.

EZZ8462I *ip_addr port_num* **IS BEING DISTRIBUTED TO ALL STACKS**

Explanation: This message is additional information for message EZZ8470I or EZZ8471I. Incoming connections to the specified IP address-port pair are currently being distributed among all stacks in the sysplex group. This message further explains why a VIPADISTRIBUTE DEFINE statement was ignored, or a VIPADISTRIBUTE DELETE statement was rejected. If the specified IP address and port were in a VIPADISTRIBUTE DEFINE statement with a specified DESTIP address, the VIPADISTRIBUTE DEFINE statement is ignored because the specified DESTIP address is already eligible for incoming connections to this IP address and port. If the specified IP address and port were in a VIPADISTRIBUTE DELETE statement with a specified DESTIP address, the VIPADISTRIBUTE DELETE statement is rejected because this IP address-port pair is currently configured for distribution to all stacks and exclusion of one stack is not supported.

ip_addr is the IP address of the Dynamic VIPA.

port_num is the distributed port.

System Action: TCP/IP continues. Distribution continues unchanged.

System Programmer Response:

- If the specified IP address or port is not correct, resubmit the VIPADISTRIBUTE statement with the correct IP address and port.

- If the statement was VIPADISTRIBUTE DEFINE, no additional action is necessary.
- If the statement was VIPADISTRIBUTE DELETE to exclude distribution to a particular stack submit:
  1. VIPADISTRIBUTE DELETE statement specifying DESTIP ALL
  2. VIPADISTRIBUTE DEFINE statements for each stack that is to be a target of this VIPADISTRIBUTE

EZZ8463I *ip_addr port_num destip* **NOT FOUND**

Explanation: This message is additional information for message EZZ8472I. A VIPADISTRIBUTE DELETE statement appears in a profile or VARY OBEY file but the specified IP address, port, and DESTIP are not currently configured for distribution by this stack.

ip_addr is the IP address of the Dynamic VIPA.

port_num is the distributed port.

destip is the DXCF IP address of the destination stack.

System Action: TCP/IP continues. The VIPADISTRIBUTE DELETE statement is ignored.

System Programmer Response: If the specified IP address, port, or DESTIP is not correct, submit a corrected VIPADISTRIBUTE statement.

EZZ8464I **VIPADIST DELETE** *ip address port* **REJECTED - VIPADIST ALL**

Explanation: A VIPADISTRIBUTE DELETE statement appears in a profile or VARY OBEY file specifying an ipaddress, port, and specific destination XCF address to be excluded from distribution. However, the specified ipaddress and port is currently configured for distribution to all stacks and exclusion of one stack is not supported. The VIPADISTRIBUTE DELETE statement is ignored.

ip address is the IP address of the Dynamic VIPA.

port is the distributed port.

System Action: TCP/IP continues.

System Programmer Response: If the specified ipaddress, port, and/or DESTIP ipaddress is not correct, resubmit a corrected VIPADISTRIBUTE statement. If the specified ipaddress, port, and DESTIP are correct and you intend to exclude the stack with that DESTIP ipaddress from distribution, you must submit a VIPADISTRIBUTE DELETE statement specifying DESTIP ALL along with individual VIPADISTRIBUTE DEFINE statements for each stack that is to be the target of this dynamic VIPA distribution.

EZZ8465I *ip_addr* **IS A DESTINATION FOR A VIPADISTRIBUTE**

Explanation: This message is additional information for message EZZ8306I. A VIPADELETE statement appears in a profile or VARY OBEY file specifying an IP address that currently exists as a destination dynamic VIPA on this stack. Destination dynamic VIPAs cannot be deleted using the VIPADELETE configuration statement.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues. The VIPADELETE statement is rejected.

System Programmer Response: If the specified IP address is not correct, submit a corrected VIPADELETE statement. To remove the specified IP address from this stack submit a VIPADISTRIBUTE DELETE statement on the stack that is distributing this dynamic VIPA. The DESTIP address on the VIPADISTRIBUTE DELETE statement must either specify this stack's Dynamic XCF IP address or the keyword 'ALL' to end distribution to all stacks for the specified IP address and port.

EZZ8466I **VIPADISTRIBUTE REJECTED - DYNAMIC XCF IS NOT ENABLED**

Explanation: A VIPADISTRIBUTE statement appears in a profile or VARY OBEY file, but Dynamic XCF is not enabled on this stack.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is ignored.

System Programmer Response: Enable Dynamic XCF with the IPCONFIG DYNAMICXCF configuration statement. Either correct and resubmit the original profile or submit a VARY TCPIP,,OBEYFILE command.

EZZ8467I VIPADISTRIBUTE WOULD EXCEED MAXIMUM DESTIPS

Explanation: This message is additional information for message EZZ8469I. A VIPADISTRIBUTE DEFINE statement appears in a profile or VARY OBEY file, but the specified IP address and port are currently configured for distribution to the maximum number of destination IP addresses.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is rejected.

System Programmer Response: Use one of the following commands to see the DESTIPs configured on this stack:

- display tcpip,tcpip_procname,netstat,vipadcfg
- onetstat -F
- netstat vipadcfg

Submit a VIPADISTRIBUTE DELETE statement followed by a VIPADISTRIBUTE DEFINE statement to remove an existing DESTIP and configure the new DESTIP.

EZZ8468I VIPADISTRIBUTE WOULD EXCEED MAXIMUM PORTS

Explanation: This message is additional information for message EZZ8469I. A VIPADISTRIBUTE DEFINE statement appears in a profile or VARY OBEY file, but the specified IP address is currently configured for distribution to the maximum number of ports.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is rejected.

System Programmer Response: Use one of the following commands to see the ports configured on this stack:

- display tcpip,tcpip_procname,netstat,vipadcfg
- onetstat -F
- netstat vipadcfg

Submit a VIPADISTRIBUTE DELETE statement followed by a VIPADISTRIBUTE DEFINE statement to remove an existing port and configure the new port.

EZZ8469I VIPADISTRIBUTE *ip_addr port_num destip* **REJECTED**

Explanation: A VIPADISTRIBUTE DEFINE statement appears in a profile or VARY OBEY file, but the specified IP address and port cannot be distributed to the specified DESTIP address. An additional message is issued to describe the reason.

ip_addr is the IP address of the Dynamic VIPA.

port_num is the distributed port.

destip is the DXCF IP address of the destination stack.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is rejected.

System Programmer Response: See the additional message.

EZZ8470I VIPADISTRIBUTE *ip_addr port_num destip* **IGNORED**

Explanation: A VIPADISTRIBUTE DEFINE statement appears in a profile or VARY OBEY file, but the specified IP address and port are already distributed to the specified DESTIP. An additional message is issued to describe the reason.

ip_addr is the IP address of the Dynamic VIPA.

port_num is the distributed port.

destip is the DXCF IP address of the destination stack.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is ignored.

System Programmer Response: See the additional message.

EZZ8471I VIPADIST DEL *ip_addr port_num destip* **REJECTED**

Explanation: A VIPADISTRIBUTE DELETE statement appears in a profile or VARY OBEY file, but distribution of the specified IP address and port to the specified DESTIP address could not be deleted. An additional message is issued to describe the reason.

ip_addr is the IP address of the Dynamic VIPA.

port_num is the distributed port.

destip is the DXCF IP address of the destination stack.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is rejected.

System Programmer Response: See the additional message.

EZZ8472I VIPADIST DEL *ip_addr port_num destip* **IGNORED**

Explanation: A VIPADISTRIBUTE DELETE statement appears in a profile or VARY OBEY file, but the specified IP address and port are not currently being distributed to the specified DESTIP address. An additional message is issued to describe the reason.

ip_addr is the IP address of the Dynamic VIPA.

port_num is the distributed port.

destip is the DXCF IP address of the destination stack.

System Action: TCP/IP continues. The VIPADISTRIBUTE statement is ignored.

System Programmer Response: See the additional message.

EZZ8473I *ip_addr* **IS NOT A DYNAMIC VIPA**

Explanation: This message is additional information for message EZZ8306I. A VIPADELETE statement appears in a profile or VARY OBEY file, but the specified IP address is not configured as a dynamic VIPA on this stack.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues. The VIPADELETE statement is rejected.

System Programmer Response: Specify a dynamic VIPA address.

EZZ8474I *ip_addr* **IS A DISTRIBUTING VIPA**

Explanation: This message is additional information for message EZZ8306I. A VIPADELETE statement appears in a profile or VARY OBEY file but the specified dynamic VIPA address is currently configured as distributing to one or more stacks in the sysplex group.

ip_addr is the IP address of the Dynamic VIPA.

System Action: TCP/IP continues. The VIPADELETE is rejected.

System Programmer Response: To delete this dynamic VIPA, you must first issue one or more VIPADISTRIBUTE DELETE statements to end all distributions for this dynamic VIPA. Use one of the following commands to see the distributions configured for this dynamic VIPA:

- display tcpip,tcpip_procname,netstat,vipadcfg
- onetstat -F
- netstat vipadcfg

Then issue the VIPADELETE statement.

EZZ8475I SYSPLEX DISTRIBUTOR SERVICE MANAGER FUNCTION NOT AVAILABLE

Explanation: The Sysplex Distributor was not able to open and bind to the UDP port specified on the SMPORT parameter on the VIPASMPARMS statement in the TCPIP profile. This probably means that this port is already reserved or that another application is already bound to this port.

System Action: The Sysplex Distributor will not perform the Service Manager function for the Cisco Multi-Node Load Balancer (MNLB).

User or Operator Response: Contact the system programmer.

System Programmer Response: Update the TCPIP profile to specify a different port on the SMPORT parameter on the VIPASMPARMS statement.

Source Data Set: TCPIP

Procedure Name: EZBXFCIN

EZZ8476I NO VIPASMPARMS STATEMENT FOR VIPA *ip_address*

Explanation: The Sysplex Distributor cannot perform the Service Manager function for the indicated VIPA because there is no VIPASMPARMS statement in the TCPIP profile.

ip_address is a VIPA specified on a VIPADISTRIBUTE statement.

System Action: The Sysplex Distributor accepts the VIPADISTRIBUTE statement but will not perform the Service Manager function without a VIPASMPARMS statement.

User or Operator Response: Contact the system programmer.

System Programmer Response: Update the TCPIP profile to specify a VIPASMPARMS statement in the VIPADYNAMIC block.

Source Data Set: TCPIP

Procedure Name: EZBXFDYN

EZZ8477I INCONSISTENT VIPASMPARMS DEFINED BY *tcp_jobname* **ON** *mvsname*

Explanation: The VIPASMPARMS statement in the TCPIP profile on this stack is inconsistent with the VIPASMPARMS statement in the TCPIP profile for another TCP/IP stack in the sysplex.

tcp_jobname is the jobname of the other TCP/IP stack.

mvsname is the name of the system on which the other TCP/IP stack is running.

System Action: The Sysplex Distributor on this stack will use the parameters defined on the VIPASMPARMS statement in the TCPIP profile. However, this inconsistent definition across the stacks probably means that the Sysplex Distributor will not provide the intended workload distribution within the sysplex for Dynamic VIPAs with the SERVICEMGR parameter.

User or Operator Response: Contact the system programmer.

System Programmer Response: Review the VIPASMPARMS statements in the respective TCPIP profiles and, if necessary, update these definitions.

Source Data Set: TCPIP

Procedure Name: EZBXFMSI

EZZ8495I TRMD STARTED

Explanation: trmd was started.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8496I trmd internal error:*indicator*

Explanation: An unexpected internal error occurred. The value of *indicator* can be one of the following:

76 An error occurred while opening a socket. Ensure that the stack is running and was initialized correctly.

77 An error occurred while forking.

78 trmd could not allocate enough storage. Increase your region size.

79 An error occurred while setting up a signal handler.

System Action: Application ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8498I TRMD STACK UNAVAILABLE:*stackname*

Explanation: trmd was unable to establish affinity to a TCP/IP stack.

System Action: Application ended.

User or Operator Response: None.

System Programmer Response: Ensure the stack was initialized correctly.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8500I TRMD INITIALIZATION COMPLETE

Explanation: trmd initialized successfully.

System Action: Application continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8501I TRMD ENDED

Explanation: trmd is ending.

System Action: Application ended.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8502I TRMD ARGUMENT INVALID

Explanation: An invalid argument was specified.

System Action: Application ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8503I TRMD ALREADY RUNNING ON: *stackname*

Explanation: A copy of trmd is already running. Only one copy of trmd may be running per TCP/IP image at a time.

System Action: Application ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8504I No value specified for the *option* **option**

Explanation: The trmdstat command was invoked with an option without a value.

option is a command line option.

System Action: The trmdstat command ends.

User or Operator Response: Correct the error and reissue the trmdstat command with the correct option value.

System Programmer Response: None.

Source Data Set: TSTATINI.C

Procedure Name: tstatParseCmdLineArgs

EZZ8505I Value specified for *option* **option is out of range**

Explanation: While processing the command line options for the trmdstat command, an option was encountered that required a numeric value within a specified range. The value specified was outside the allowable range of values for the option being processed.

option is a command line option.

System Action: The trmdstat command ends.

User or Operator Response: Correct the value specified for the option in error and reissue the trmdstat command.

System Programmer Response: None.

Source Data Set: TSTATINI.C

Procedure Name: tstatParseCmdLineArgs

EZZ8506I Value specified for the *option* **option is not numeric**

Explanation: While processing the command line options for the trmdstat command, an option was encountered that required a numeric value. The value specified was not numeric.

option is a command line option.

System Action: The trmdstat command ends.

User or Operator Response: Correct the value specified for the option in error and reissue the trmdstat command.

System Programmer Response: None.

Source Data Set: TSTATINI.C

Procedure Name: tstatParseCmdLineArgs

EZZ8507I IP address *address* **specified for the -h option is not in the correct format**

Explanation: The trmdstat command was unable to interpret the IP address correctly.

address is the IP address.

System Action: The trmdstat command ends.

User or Operator Response: Correct the value specified for the IP address and reissue the trmdstat command.

System Programmer Response: None.

Source Data Set: TSTATINI.C

Procedure Name: tstatParseCmdLineArgs

EZZ8508I Unable to open *input_file* **-** *error_text*

Explanation: The trmdstat command was unable to open the *input_file* specified in the command line.

input_file is the name of the input file.

error_text is the error message string, containing a specific reason for the failure.

System Action: The trmdstat command ends.

User or Operator Response: Correct the error and reissue the trmdstat command.

System Programmer Response: None.

Source Data Set: TSTATMN.C

Procedure Name: main

EZZ8509I Unable to allocate memory in *function*

Explanation: The trmdstat command attempted to allocate memory in function *function*. Memory could not be obtained.

function is the name of the function that cannot allocate memory.

System Action: The trmdstat command ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Determine why memory was not available. Correct the problem and reissue the trmdstat command.

Source Data Set: TSTATLST.C, TSTATPRO.C

Procedure Name: tstatUpdateConnList, tstatUpdateDetInfo, tstatUpdateConnHostInfo, tstatUpdateTermMsgs

---

EZZ8510I Unable to open message catalog *catalog_name* **-** *error_text*

Explanation: The trmdstat command was unable to open the message catalog *trmdstat.cat* in the message catalog directory. The default location for the message catalog is set by the NLSPATH environment variable to be NLSPATH=/usr/lib/nls/msg/%L/%N.

catalog_name is the name of the message catalog.

error_text is the error message string, containing a more specific reason for the failure.

System Action: The trmdstat command will use the internal default messages instead of the messages from the external message catalog.

User or Operator Response: If you want to use the external message catalog, contact the system programmer. If the default messages are acceptable, no action is necessary.

System Programmer Response: If you want to use the external message catalog, correct the indicated error. There are several reasons that could cause this error, such as file or directory permissions not allowing read access. Refer to the *z/OS C/C++ Run-Time Library Reference* for more information about the catopen() function call. Refer to the *z/OS*

UNIX System Services Programming Tools for more information about the NLSPATH environment variables. If the default messages are acceptable, no action is necessary.

Source Data Set: TSTATINI.C

Procedure Name: tstatInitialize

EZZ8511I Usage: trmdstat <Options> log_file_name

Explanation: The following is the complete text of the message:

```
EZZ8511I Usage: trmdstat <Options> log_file_name
Report type options:
  -A              - Displays the ATTACK report
  -C              - Displays the CONNECTION report
  -F              - Displays the FLOOD report
  -I              - Displays the IDS report
  -N              - Displays the SCAN report
  -T              - Displays the TCP TR report
  -U              - Displays the UDP TR report
  default         - Displays the TCP TR summary report
Report content options:
  -D              - For -A, -C, -F, -N, -T or -U
                    displays the details information
  -E              - For -T
                    displays the extended summary information
  -S              - For -A, -T, or -U
                    displays statistics information
Stack name option:
     -j stack name   - The name of the TCP/IP stack for which records
                       are to be included
IP Address and port range options:
  -h ip_address   - For -A, -C, -F, -N or -U
                    displays information for ip_address
  -k ip_address   - For -T and -S
                    displays information for the peak ip_address
  -s ip_address   - For -A or -T
                    displays information for the source ip_address
  -t ip_address   - For -A or -T
                    displays information for the destination
                    ip_address
  -p port_range   - Port range to be included
Time range options:
  -i initial_time - Time of the first record to be included
  -f final_time   - Time of the final record to be included
Debug option:
  -d debug_level  - Specifies the debug level
```

TRMDSTAT was invoked requesting help or with a specification error.

System Action: TRMDSTAT processing ends.

User or Operator Response: Restart TRMDSTAT.

System Programmer Response: None.

Source Data Set: TRMDSTAT

Procedure Name: main

EZZ8540I SUPERUSER AUTHORITY REQUIRED

Explanation: TRMD was invoked from a user ID that was not a superuser.

System Action: TRMD processing ends.

User or Operator Response: Restart TRMD from a superuser.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: main

EZZ8600I DCAS UNABLE TO OPEN MESSAGE CATALOG dcasm.cat ERRNO *errno*

Explanation: The Digital Certificate Access Server was unable to open the dcasm.cat message catalog. The default location for the message catalog is set by the NLSPATH environment variable to be ″NLSPATH=/usr/lib/nls/msg/%L/%N″.

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

System Action: Initialization continues.

User or Operator Response: None.

System Programmer Response: If you want to use the external message catalog, correct the indicated error. If the default messages are acceptable, no action is necessary.

Source Data Set: dcasmain

Procedure Name: main

EZZ8601I DCAS IS STARTING

Explanation: The Digital Certificate Access Server is initializing.

System Action: Initialization continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasmain

Procedure Name: main

EZZ8602I DCAS ENDED

Explanation: The Digital Certificate Access Server ended.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasmain

Procedure Name: termprocess

EZZ8603I DCAS INVALID CLIENTAUTH VALUE

Explanation: The CLIENTAUTH keyword value specified in the DCAS configuration file is incorrect.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Correct the CLIENTAUTH value and restart DCAS.

Source Data Set: dcasconf

Procedure Name: process_passtick_config

EZZ8604I DCAS IS ATTEMPTING TO END ALL THREADS

Explanation: This indicates that the DCAS ending process is starting.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasmain

Procedure Name: termthreads

EZZ8605I DCAS INITIAL VALIDATIONS FAILED: *errtype*

Explanation: When trying to start DCAS, some validation conditions were not met.

errtype can be one of the following:

not a superuser
DCAS must run with superuser authority.

not an authorized application
DCAS must run as an authorized application.

not authorized to start
User is not authorized to start DCAS.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Review the following:

not a superuser
Change DCAS to run with superuser authority.

not an authorized application
Change DCAS to run from an APF authorized library.

not authorized to start
Use RACF to define the user ID under which DCAS is started to the MVS.SERVMGR.DCAS profile in the OPERCMDS class. Do the following RACF commands:

- Rdefine OPERCMDS MVS.SERVMGR.DCAS UACC(NONE)
- Permit MVS.SERVMGR.DCAS CLASS(OPERCMDS) ACCESS(CONTROL) ID(userid)

Source Data Set: dcasinit

Procedure Name: validations

EZZ8606I DCAS INITIALIZATIONS FAILED: *errtype ERRNO errno*

Explanation: When trying to start DCAS, a main thread initialization failed.

errtype can be one of the following:

Thread Attribute
Failed creating a thread attribute.

Thread Key
Failed creating a thread key.

errno *errno* is the decimal UNIX System Services Return Code set by the LE C Run-Time service that DCAS invoked. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*. The LE Run-Time function invocation that failed is one of the following:

Thread Attribute, Thread Key
pthread_attr_init()

Thread Key
pthread_key_create()

System Action: DCAS is not started.

User or Operator Response: None.

System Programmer Response: Make sure that DCAS is started with the POSIX(ON) runtime option. DCAS uses the LE C/C++ Run-Time library services. Verify that a problem does not exist with the LE Run-Time Library. The *z/OS C/C++ Run-Time Library Reference* describes the function calls and the errno's returned.

Source Data Set: dcasmain dcasinit

Procedure Name: initprocess daemonize

EZZ8607I DCAS *threaddesc* **THREAD** *service* **FAILED ERRNO** *errno*

Explanation: The LE C/C++ Run-Time service that failed is one of the following:

threaddesc
Type of thread where failure occurred can be one of the following:

MAIN DCAS main thread.

PORT DCAS port thread.

STOP DCAS stop thread.

CNxxxxxx
DCAS client thread where xxxxxx is a number concatenated to CN to identify the client thread uniquely.

service The runtime service that was performed failed and can be one of the following:

SETINTRTYPE
Thread invocation of pthread_setintrtype() returned an error

GETSPECIFIC
Thread invocation of pthread_getspecific() returned an error

SETSPECIFIC
Thread invocation of pthread_setspecific() returned an error

CREATE
Thread invocation of pthread_create() returned an error

errno *errno* is the decimal UNIX System Services Return Code set by the LE C Run-Time service that DCAS invoked. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*. The LE Run-Time function invocation that failed is one of the following:

System Action: The thread where the failure occurred will terminate. If the *threaddesc* is MAIN, PORT, or STOP, DCAS ends. If the *threaddesc* is CNxxxxxx, the client thread ends.

User or Operator Response: None.

System Programmer Response: Make sure that DCAS is started with the POSIX(ON) runtime option. DCAS uses the LE C/C++ Run-Time library services. Verify that a problem does not exist with the LE Run-Time Library. The *z/OS C/C++ Run-Time Library Reference* describes the function calls and the errno's returned.

Source Data Set:

Procedure Name: main

EZZ8608I DCAS *threaddesc* **THREAD FAILED OBTAINING STORAGE LENGTH** *storlen* **FOR** *storagedesc*

Explanation: DCAS was unable to obtain storage.

threaddesc indicates the type of thread that had the failure and can be one of the following:

MAIN DCAS main thread.

PORT DCAS port thread.

EZZ8609I

CNxxxxxx
DCAS client thread where xxxxxx is a number concatenated to CN to identify the client thread uniquely.

storlen is the length of storage in decimal that the thread tried to obtain.

storagedesc is the representation of storage that the thread was trying to obtain. These are internal data structures that might be useful for problem determination. *storagedesc* can be one of the following:

- DCAS_Main
- Configuration Parm
- Keyring Filename
- Password Filename
- SafKeyring Filename
- Port
- Thread Data
- Certificate
- Conntab
- Connblk

System Action: Thread will fail. For *threaddesc* of MAIN, PORT, or STOP, DCAS will end.

User or Operator Response: None.

System Programmer Response: Increase the region size for the started program. If the problem persists, turn on debugging and capture a debug trace. Refer to the *z/OS Communications Server: IP Diagnosis* for information about diagnosing problems with DCAS to determine how to turn on debugging.

Source Data Set: dcasmain dcasconf dcasaccp dcasconn dcasclie

Procedure Name: Initprocess; process_config_keyword,initport; acceptcon; connt_alloc,connblk_alloc;ClientThread

---

EZZ8609I DCAS SIGNAL FUNCTION *sigfunc* **FAILED FOR** *signal* **ERRNO** *errno*

Explanation: DCAS performed a signal function using the LE C Run-Time services, and the signal function returned an error.

sigfunc is the signal function that failed can be one of the following:

sigaddset()
Add a signal to a signal set.

sigwait()
Wait on a set of signals.

sigaction()
Perform an action when signal occurs.

signal is the signal that failed.For *sigfunc* of sigwait(), this is NA (not applicable). These are OS/390 Unix system signals. They are described in the *z/OS C/C++ Programming Guide*.

errno is the decimal UNIX System Services Return Code set by the LE C Run-Time service that DCAS invoked. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

System Action: For *sigfunc* of sigwait(), DCAS will end; otherwise, DCAS will terminate the thread and cause DCAS to end if the thread was the listening port thread.

User or Operator Response: None.

System Programmer Response: Make sure that DCAS is started with the POSIX(ON) runtime option. DCAS uses the LE C/C++ Run-Time library services. Verify that a problem does not exist with the LE Run-Time Library. The *z/OS C/C++ Run-Time Library Reference* describes the function calls and the errno's returned.

Source Data Set: dcasmain dcasaccp dcasclie

Procedure Name: main,main_signals;accp_signals;clie_signals

EZZ8610I DCAS OPEN FOR FILE *filename* **FAILED ERRNO:** *errno*

Explanation: Open for the file failed.

filename is the name of the file that failed to open.

errno is the decimal UNIX System Services Return Code set by the LE C Run-Time service that DCAS invoked. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Check if the file exists. HFS filenames are case sensitive. Correct the problem and restart DCAS.

Source Data Set: processopts Demonize

Procedure Name: dcasmain dcasinit

EZZ8611I DCAS failed to establish affinity to *tcpipname* **ERROR** *errno/errnojr*

Explanation: DCAS tried to establish affinity to a TCP/IP stack name specified in the DCAS configuration. The name used is the name specified with the TCP/IP keyword in the DCAS configuration file. If DEFAULT was specified, then the TCP/IP jobname from the TCPIP.DATA file is used.

tcpipname is the name used to establish affinity to.

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

errnojr is the hexadecimal UNIX System Services Reason Code returned by the setibmopt(). The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Name specified is not configured to be a valid socket PFS. Make sure that it is a valid TCP/IP jobname.

Source Data Set:

Procedure Name:

EZZ8612I DCAS *sockfunc* **Call FAILED** *errno/errnojr* **FOR PORT** *portno*

Explanation: One of the initial socket calls to activate the port failed.

sockfunc is the socket call that failed.

errno is the hexidecimal UNIX System Services Return Code returned by the *sockfunc*. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

errnojr is the hexadecimal UNIX System Services Reason Code returned by the *sockfunc*. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: The errno/errnojr give the reason for the *sockfunc* failure.

Source Data Set: dcassock

Procedure Name: Initsock

EZZ8613I DCAS SSL INITIALIZATION FAILED RC:*rcode* **ERRNO:** *errno*

Explanation: DCAS attempt to initialize Secure Socket Layer (SSL) functions failed.

rcode is the return code from gsk_initialize() and can be one of the following:

4 Password for keyring file is incorrect

8 The keyring file is incorrect

102 The sec_types value was incorrect

103 The V2_session timeout value was incorrect

104 The V3_session timeout value was incorrect

-10 GSK_IO_ERROR

errno is the UNIX System Services Return Code. These return codes are listed and described in the *z/OS UNIX System Services Messages and Codes*. The errno is returned only if the return code is GSK_IO_ERROR. The translated *errno* is one of the following:

EINVAL
Parameter is not valid or null on gsk_initialize()

EFAULT
Invalid address on gsk_initialize()

EUKNOWN
Unknown error occurred

System Action: DCAS ended.

User or Operator Response: None.

System Programmer Response: Determine the reason for the failure from the return code and errno. Correct the problem and restart the DCAS. The failure is most likely due to a problem with the KEYRING or STASHFILE file or the SAFKEYRING value that was specified in the DCAS configuration.

Source Data Set: dcasgsk

Procedure Name: gskinit

EZZ8614I DCAS SSL OBTAIN CIPHER FAILED RC:*rcode* **ERRNO:** *errno*

Explanation: Attempt to obtain cipher information from SSL failed.

rcode is the return code from the gsk_get_cipher_info()

-10 Indicates GSK_IO_ERROR for SSL.

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

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Determine the reason for the failure from the return code and errno. Correct the problem and restart the DCAS.

Source Data Set: dcasgsk

Procedure Name: gskinit

EZZ8615I DCAS NOTIFIED THAT PORT *portnum* **HAS ENDED**

Explanation: DCAS port thread notified the main thread that the listening port and its connections have ended.

portnum is the listening port number.

System Action: Processing continues.

User or Operator Response: none

System Programmer Response: Other messages might be present if the port ended due to an error with DCAS. This message is also issued in response to a stop type command. Refer to the *z/OS Communications Server: IP Diagnosis* for information about diagnosing problems with DCAS if an error occurred.

Source Data Set: dcasmain

Procedure Name: termthreads

EZZ8616I DCAS SECURITY SERVER IS UNAVAILABLE SAFRC: *src* **RACFRC** *rrc* **RACFRSN** *rrsn*

Explanation: Security server (RACF) is not available. The return codes provided indicate the reason for the failure. The following are the codes returned by SAF and RACF in response to a RACROUTE STAT macro issued by DCAS:

src is the SAF return code.

rrc is the RACF return code.

rrsn is the RACF reason code.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: RACF (or other SAF compliant security server that provides the same function) must be active. Return codes from RACROUTE STAT macro are defined in the *z/OS Security Server RACROUTE Macro Reference*.

Source Data Set: dcasinit

Procedure Name: Racfchecks

EZZ8617I DCAS *cmdtype* **COMMAND RECEIVED**

Explanation: Operator command issued to DCAS.

cmdtype can be one of the following:

STOP DCAS ends.

MODIFY
Debugging toggle.

UNSUPPORTED
DCAS does not support this command.

System Action: For a STOP command, DCAS ends. For a MODIFY command DCAS will toggle debug level 3 to the logfile originally specified at startup or if none specified, then it will log to syslogd. For an UNSUPPORTED command, DCAS ignores the command.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasstop

Procedure Name: monitorstop

EZZ8618I DCAS LISTENING ON *porttype* **PORT** *portnum*

Explanation: DCAS port is active and accepting connections.

porttype is the type of port and can be:

SECURE
Connection on this port uses SSL for secure communication

portnum is the port number of the listening port

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasaccp

Procedure Name: acceptcon

EZZ8619I DCAS PORT THREAD IS ENDING FOR PORT *portnum*

Explanation: DCAS listening port is ending.

portnum is the port number of the listening port.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Stop DCAS and restart. Other messages might be present if the port is ending due to an error. Refer to the *z/OS Communications Server: IP Diagnosis* for information about diagnosing DCAS. The message also appears when the port ends normally.

Source Data Set: dcasaccp

Procedure Name: acceptcon

EZZ8620I DCAS security server SERVAUTH class is *status*

Explanation: *status* is the status of the RACF SERVAUTH class and can be one of the following:

active SERVAUTH class is active.

inactive
SERVAUTH class is inactive.

undefined
SERVAUTH class is undefined.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: If DCAS is configured with CLIENTAUTH LOCAL2 and extended client authentication using the SERVAUTH class is desired, then the SERVAUTH must be active. Using the RACF RDEFINE command, define the SERVAUTH CLASS.

Source Data Set: dcasinit

Procedure Name: Racfchecks

EZZ8621I DCAS SECURITY SERVER STATUS NOT DETERMINED

Explanation: DCAS checked the status of the security server (RACF). The return code from the security server indicated that the status of RACF could not be determined.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: RACF router was not loaded; the request, resource, subsystem combination could not be found in the RACF ROUTER table. Successful RACF exit processing could not take place. Verify RACF installation and setup.

Source Data Set: dcasinit

Procedure Name: Racfchecks

EZZ8622I DCAS CONFIGURATION PROCESSED UNKNOWN KEYWORD *keyword*

Explanation: DCAS configuration file processor found an unknown keyword.

keyword is the unknown keyword processed.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: Correct the keyword in the configuration file. Stop and restart DCAS.

Source Data Set: dcasconf

Procedure Name: process_keyword

EZZ8623I DCAS CONFIGURATION FILE KEYWORD *keyword* **DOES NOT HAVE A VALUE**

Explanation: DCAS configuration file processing found a valid keyword that does not have a value.

keyword is the keyword processed.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: Update the keyword indicated in the DCAS configuration file and restart DCAS.

Source Data Set: dcasconf

Procedure Name: process_keyword

EZZ8624I DCAS PROCESSING CONFIGURATION FILE *filename*

Explanation: DCAS is processing the configuration file.

filename is the filename processed.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasmain

Procedure Name: main

EZZ8625I DCAS CONFIGURATION FILE PROCESSING IS COMPLETE

Explanation: Configuration file was processed.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: dcasmain

Procedure Name: main

EZZ8626I DCAS PORT *portnum***SIGNAL ACTION** *signal* **OCCURRED**

Explanation: Signal action occurred on the port thread.

portnum is the port number associated with the listening port thread.

signal is the signal that occurred and can be one of the following:

SIGABND Abend.

SIGILL Invalid object module.

SIGPIPE Write on pipe with no one to read.

SIGSEGV Segmentation violation.

SIGTERM Termination signal received.

UNKNOWN Signal received that was unexpected.

System Action: For SIGABND, SIGSEGV, or SIGILL the port will attempt to recover. For SIGTERM or SIGPIPE the port will terminate. A CEEDUMP may have been taken.

User or Operator Response: None.

System Programmer Response: Save CEEDUMP for diagnosis if one was taken. Signals refer to z/OS Unix system signals which are described in the *z/OS C/C++ Programming Guide*.

Source Data Set: dcasaccp

Procedure Name: acceptcon

EZZ8627I DCAS *clientid* **SIGNAL ACTION** *signal* **OCCURRED**

Explanation: Client thread received a signal.

signal is the signal that occurred and can be one of the following:

SIGABND Abend.

SIGILL Invalid object module.

SIGSEGV Segmentation violation.

SIGPIPE Write on pipe with no one to read.

SIGTERM Termination signal received.

UNKNOWN Signal received that was unexpected.

System Action: Client thread ends. A CEEDUMP may have been taken.

User or Operator Response: None.

System Programmer Response: Save CEEDUMP for diagnosis if one was taken. Signals refer to z/OS Unix system signals which are described in the *z/OS C/C++ Programming Guide*.

Source Data Set: dcasclie

Procedure Name: clie_sighand

EZZ8628I DCAS DEBUG REQUESTED BUT NO LOGFILE WAS DEFINED

Explanation: Did not specify a logfile.

System Action: DCAS continues without debugging.

User or Operator Response: None.

System Programmer Response: Refer to the *z/OS Communications Server: IP Configuration Guide* to see how to specify a log file. Stop and restart DCAS with logging.

Source Data Set: dcasmain

Procedure Name: main

EZZ8629I DCAS SAFKEYRING AND KEYRING ARE MUTUALLY EXCLUSIVE - SAFKEYRING USED

Explanation: In the DCAS configuration file SAFKEYRING and KEYRING keywords were specified. They are mutually exclusive. SAFKEYRING value is used.

System Action: DCAS continues and SAFKEYRING value is used.

User or Operator Response: None.

System Programmer Response: Refer to the *z/OS Communications Server: IP Configuration Reference* for information about configuring the DCAS.

Source Data Set: dcasconf

Procedure Name: process_passtick_config

EZZ8630I DCAS CLIENTAUTH DEFAULTED TO LOCAL2

Explanation: CLIENTAUTH keyword was not specified in the configuration file. It defaults to CLIENTAUTH LOCAL2.

System Action: DCAS continues.

User or Operator Response: None.

System Programmer Response: Refer to the *z/OS Communications Server: IP Configuration Reference* for information about configuring the DCAS and specifying the CLIENTAUTH keyword.

Source Data Set: dcasconf

Procedure Name: process_passtick_config

EZZ8631I DCAS NO CONFIGURATION FILE

Explanation: A DCAS configuration file was not specified and no default configuration file was found.

System Action: DCAS ends.

User or Operator Response: None.

System Programmer Response: Start the DCAS with a valid configuration file. Refer to the *z/OS Communications Server: IP Configuration Reference* for information about specifying a DCAS configuration file.

Source Data Set: dcasmain

Procedure Name: main

EZZ8638I TRMD UDP constrained entry logged: *timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **qsize=***qsize*, **correlator=***correlator*

Explanation: The Intrusion Detection Services (IDS) policy for UDP did not specify TypeActions Limit. The inbound queue exceeds 90% of the policy defined queue size.

timestamp is the date and time the inbound queue for a UDP port became constrained.

dipaddr is the destination IP address that triggered the storage constraint.

dport is the destination port specified in the policy.

qsize is the queue size specified in the policy.

correlator is the IDS trace correlator.

System Action: Processing continues

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8639I TRMD UDP constrained entry:*timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **qsize=***qsize*, **correlator=***correlator*

Explanation: The Intrusion Detection Services (IDS) policy for UDP specified TypeActions Limit. The inbound queue exceeds 90% of the policy defined queue size. No more datagrams will be accepted until this condition is alleviated.

timestamp is the date and time the inbound queue for a UDP port became constrained.

dipaddr is the destination IP address that triggered the storage constraint.

dport is the destination port specified in the policy.

qsize is the queue size specified in the policy.

correlator is the IDS trace correlator.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8640I TRMD UDP constrained exit logged:*timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **qsize=***qsize*, **correlator=***correlator*, **datagrams=***datagrams*, **duration=***duration*

Explanation: The Intrusion Detection Services (IDS) policy for UDP did not specify TypeActions Limit. The inbound queue was reduced below 80% of the policy defined queue size.

timestamp is the date and time the inbound queue for a UDP port exited constraint.

dipaddr is the destination IP address that triggered the storage constraint.

dport is the destination port specified in the policy.

qsize is the queue size specified in the policy.

correlator is the Intrusion Detection Services (IDS) trace correlator.

datagrams is the number of datagrams that would have been discarded during the constraint period.

duration is the number of seconds the inbound queue was constrained.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8641I TRMD UDP constrained exit:*timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **qsize=***qsize*, **correlator=***correlator*, **datagrams=***datagrams*, **duration=***duration*

Explanation: The Intrusion Detection Services (IDS) policy for UDP specified TypeActions Limit. The inbound queue was reduced below 80% of the policy defined queue size.

timestamp is the date and time the inbound queue for a UDP port exited constraint.

dipaddr is the destination IP address that triggered the storage constraint.

dport is the destination port specified in the policy.

qsize is the queue size specified in the policy.

correlator is the IDS trace correlator.

datagrams is the number of datagrams that were discarded during the constraint period.

duration is the number of seconds the queue was constrained.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8642I **TRMD UDP statistics:***timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **qsize=***qsize*, **dagsrecv=***dagsrecv*, **dagsdisc=***dagsdisc*, **bytsrecv=***bytsrecv*, **bytsdisc=***bytsdisc*, **bytes_peak=***bytes_peak*, **dg_peak=***dg_peak*, **duration=***duration*, **constraints=***constraints*, **action=***action*

Explanation: The Intrusion Detection Services (IDS) UDP statistics have been gathered.

timestamp is the date and time the UDP statistics were gathered.

dipaddr is the destination IP address.

dport is the bound port specified in the policy.

qsize is the queue size specified in the policy.

dagsrecv is the number of datagrams received. If *action* is NOLIMIT then this includes the number of datagrams that would have been discarded as shown in *dagsdisc*. If *action* is LIMIT then this does not include the number of datagrams that were discarded as shown in *dagsdisc*.

dagsdisc is the number of datagrams discarded (if *action* is LIMIT) or the number of datagrams that would have been discarded (if *action* is NOLIMIT.)

bytsrecv is the number of bytes received. If *action* is NOLIMIT then this includes the number of bytes that would have been discarded as shown in *bytsdisc*. If *action* is LIMIT then this does not include the number of bytes that were discarded as shown in *bytsdisc*.

bytsdisc is the number of bytes discarded (if *action* is LIMIT) or the number of bytes that would have been discarded (if *action* is NOLIMIT.).

bytes_peak is the largest number of bytes queued during the statistics interval. This field is set only if a receive was processed during the statistics interval.

dg_peak is the largest number of datagrams queued during the statistics interval. This field is set only if a receive was processed during the statistics interval. Datagrams from a Pascal API are not included in the count.

duration is the number of seconds UDP inbound queue was constrained.

constraints is the number of times the constrained state was entered.

action is the policy TypeActions indicator. LIMIT indicates TypeActions LIMIT was specified in the policy; NOLIMIT indicates TypeActions LIMIT was not specified in the policy.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteStatEntries

EZZ8643I **TRMD SCAN threshold exceeded:***timestamp*, **sipaddr=***sipaddr*, **scantype=***scantype*, **pthreshold=***pthreshold*, **pinterval=***pinterval*, **vs=***vs*, **ps=***ps*, **norm=***norm*, **correlator=***correlator*

Explanation: A possible fast or slow scan was detected from a source IP address.

timestamp is the date and time the scan event was detected.

sipaddr is the source IP address that triggered the scan detection.

scantype is the type of scan experienced. F indicates a fast scan; S indicates a slow scan.

pthreshold is the fast or slow scan threshold specified in the policy.

pinterval is the scan interval specified in the policy.

vs is the number of very suspicious events encountered before reaching the threshold. See the *z/OS Communications Server: IP Configuration Guide* for a description of very suspicious events.

ps is the number of possibly suspicious events encountered before reaching the threshold. See the *z/OS Communications Server: IP Configuration Guide* for a description of possibly suspicious events.

norm is the number of normal events encountered before reaching the threshold. See the *z/OS Communications*

Server: IP Configuration Guide for a description of normal events.

correlator is the Intrusion Detection Services (IDS) trace correlator.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8644I TRMD SCAN detail:*timestamp*, **sipaddr=***sipaddr*, **correlator=***correlator*, **event_count=***event_count*, **event_list=***event_list*

Explanation: The Intrusion Detection Services (IDS) policy for scan specified Notification SysLogDetail. A possible fast or slow scan was reported; a summary of the events that contributed to the detection are listed.

timestamp is the date and time the scan event was detected.

sipaddr is the source IP address that triggered the scan detection.

correlator is the IDS trace correlator.

event_count is the number of events included in the following event list.

event_list is the summary information about the events that contributed to the scan. Each entry contains 4 elements:

- Destination IP address.
- Protocol
- Destination port if the protocol is UDP or TCP or ICMP. Type if the protocol is ICMP.
- Suspicion level. The suspicion level value might be V (very suspicious), P (possibly suspicious) or N (normal).

The events related to the scan might span several log records.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8645I TRMD SCAN detection delayed:*timestamp*, **correlator=***correlator*

Explanation: Scan interval processing took more than an interval to complete. Scan detection logging might be delayed.

timestamp is the date and time the scan event was detected.

correlator is the Intrusion Detection Trace (IDS) trace correlator.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Scan processing is not able to complete its evaluation of the source ip addresses it is tracking in its normal internal interval (30 or 60 seconds). This might indicate that a large number of source ip addresses are being monitored. If the policy is using High scan sensitivity, the installation may want to consider lowering the scan sensitivity level for high usage ports.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8646I TRMD SCAN storage constrained:*timestamp*, **correlator=***correlator*

Explanation: Private storage could not be obtained to track either a source IP address or a scan event. Scan tracking may be incomplete.

timestamp is the date and time the scan event was detected.

correlator is the Intrusion Detection Services (IDS) trace correlator.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Determine the cause of the storage shortage. See the *z/OS Communications Server: IP Diagnosis* for information about storage shortages.

Source Data Set: WriteLogEntries

Procedure Name: EZATRMD

EZZ8647I TRMD SCAN storage constraint exited:*timestamp*, **correlator=***correlator*

Explanation: Scan private storage allocation failures no longer occurring. Normal Scan tracking resumed.

timestamp is the date and time the scan event was detected.

correlator is the Intrusion Detection Services (IDS) trace correlator.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8648I TRMD ATTACK packet was discarded:*timestamp*, **sipaddr=***sipaddr*, **dipaddr=***dipaddr*, **sport=***sport*, **dport=***dport*, **type=***type*, **proto=***proto*, **option=***option*, **fragsize=***fragsize*, **correlator=***correlator*, **probeid=***probeid*

Explanation: The Intrusion Detection Services (IDS) policy for ATTACK specified TypeActions Limit. A packet was discarded after an attack event of the type indicated by *type* occurred.

timestamp is the date and time the attack event occurred.

sipaddr is the source IP address.

dipaddr is the destination IP address.

sport is the source port on which the attack event occurred. A value of zero indicates that the source port was not known at the point that the attack was detected.

dport is the destination port on which the attack event occurred. A value of zero indicates that the destination port was not known at the point that the attack was detected.

type is the attack event type. It will have one of the following values:

- Malformed
- OutboundRaw
- IPFragment
- ICMP
- IPOPT
- IPPROTO
- Flood
- PerpEcho

These correspond to the AttackType values specified in IDS policy. See the *z/OS Communications Server: IP Configuration Guide* for a description of the attack types.

proto is the IP protocol type.

option is the IP option. *option* is only applicable when the *type* is IPOPT.

fragsize is the number of bytes in the fragment involved in the attack. *fragsize* is only applicable when *type* is IPFragment.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point. See the *z/OS Communications Server: IP and SNA Codes* for a description of the Intrusion Detection Services probeids.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8649I TRMD ATTACK packet would have been discarded:*timestamp*, **sipaddr=***sipaddr*, **dipaddr=***dipaddr*, **sport=***sport*, **dport=***dport*, **type=***type*, **proto=***proto*, **option=***option*, **fragsize=***fragsize*, **correlator=***correlator*, **probeid=***probeid*

Explanation: The Intrusion Detection Services (IDS) policy for ATTACK did not specify TypeActions Limit. A packet would have been discarded if TypeActions Limit had been specified after an attack event of the type indicated by *type* occurred.

timestamp is the date and time the attack event occurred.

sipaddr is the source IP address.

dipaddr is the destination IP address.

sport is the source port on which the attack event occurred. A value of zero indicates that the source port was not known at the point that the attack was detected.

dport is the destination port on which the attack event occurred. A value of zero indicates that the destination port was not known at the point that the attack was detected.

type is the attack event type. It will have one of the following values:

- Malformed
- OutboundRaw
- IPFragment
- ICMP
- IPOPT
- IPPROTO
- Flood
- PerpEcho

These correspond to the AttackType values specified in IDS policy. See the *z/OS Communications Server: IP Configuration Guide* for a description of the attack types.

proto is the IP protocol type.

option is the IP option. *option* is only applicable when the *type* is IPOPT.

fragsize is the number of bytes in the fragment involved in the attack. *fragsize* is only applicable when *type* is IPFragment.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point. See the *z/OS Communications Server: IP and SNA Codes* for a description of the Intrusion Detection Services probeids.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8650I **EZZ8650I TRMD ATTACK SYN flood start:***timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **proto=***proto*, **correlator=***correlator*

Explanation: SYN flood start implies the server is under SYN flood attack.

timestamp is the date and time the SYN flood attack started.

dipaddr is the bound IP address of the SYN flood target.

dport is the bound port of the SYN flood target.

proto is the IP protocol type.

correlator is the Intrusion Detection Services (IDS) trace correlator. You can use the correlator to find the corresponding EZZ8651I Syn Flood End message.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8651I **TRMD ATTACK SYN flood end:***timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **proto=***proto*, **correlator=***correlator*, **duration=***duration*, **synrecvd=***synrecvd*, **firstack=***firstack*, **syndiscard=***syndiscard*, **syntimeout=***syntimeout*

Explanation: SYN flood end indicates the SYN flood attack ended.

timestamp is the date and time the SYN flood attack ended.

dipaddr is the bound IP address of the SYN flood target.

dport is the bound port of the SYN flood target.

proto is the IP protocol type.

correlator is the Intrustion Detection System (IDS) trace correlator. You can use the correlator to find the corresponding EZZ8651I Syn Flood Start message.

duration is the number of seconds of the SYN flood attack.

synrecvd is the number of handshakes started during SYN flood.

firstack is the number of handshakes completed during SYN flood.

syndiscard is the number of SYNs randomly discarded during SYN flood.

syntimeout is the number of SYNs timing out during SYN flood.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8652I TRMD ATTACK accept queue expanded:*timestamp*, **dipaddr=***dipaddr*, **dport=***dport*, **proto=***proto*, **correlator=***correlator*

Explanation: The Accept queue (the queue of incomplete connections for a port) was expanded.

timestamp is the date and time the accept queue was expanded.

dipaddr is the bound IP address for which the accept queue became full.

dport is the bound port for which the accept queue became full.

proto is the IP protocol type.

correlator is the Intrusion Detection Services (IDS) trace correlator.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ8653I TRMD ATTACK statistics:*timestamp*, **type=***type*, **attacks=***attacks*, **action=***action*

Explanation: Intrusion Detection Services (IDS) ATTACK statistics have been gathered.

timestamp is the date and time statistics gathering occurred.

type is the attack event type. It will have one of the following values:

- Malformed
- OutboundRaw
- IPFragment
- ICMP
- IPOPT
- IPPROTO
- Flood
- PerpEcho

These correspond to the AttackType values specified in IDS policy. See the *z/OS Communications Server: IP Configuration Guide* for a description of the attack types.

attacks is the number of attacks of type.

action is the policy TypeActions indicator. LIMIT indicates TypeActions LIMIT was specified in the policy; NOLIMIT indicates TypeActions LIMIT was specified in the policy.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteStatEntries

EZZ8761I IDS EVENT *event_type*

Explanation: This is the first message of a message group. A complete description of the message group follows:

```
 EZZ8761I IDS EVENT DETECTED
 EZZ8762I IDS EVENT TYPE event_type
 EZZ8763I CORRELATOR  correlator - PROBEID  probe_id
[EZZ8764I SOURCE IP ADDRESS source - SOURCE PORT port]
[EZZ8765I DESTINATION IP ADDRESS dest - PORT port]
 EZZ8766I IDS RULE rule_name
 EZZ8767I IDS ACTION action_name
```

EZZ8761I

The policy-based intrusion detection system (IDS) detected an event that specified that the console operator was to be alerted. The occurrence of this message can indicate that the TCP/IP stack or a particular application is under stress. The stress might be caused by a peak in workload or might be caused by malicious activity such as packet flooding, port scanning or malformed packets.

EZZ8762I

This message provides the event type of the IDS event.

event_type can be one of the following:

TCP TOTAL CONNECTION LIMIT REACHED
TCP SOURCE IP CONNECTION LIMIT REACHED
TCP PORT CONSTRAINED
TCP PORT UNCONSTRAINED
UDP PORT QUEUE CONSTRAINED
UDP PORT QUEUE UNCONSTRAINED
FAST SCAN DETECTED
SLOW SCAN DETECTED
SCAN INTERVAL OVERRUN
SCAN STORAGE CONSTRAINED
SCAN STORAGE UNCONSTRAINED
SUSPICIOUS PACKET RECEIVED
SYN FLOOD STARTED
SYN FLOOD ENDED
ACCEPT QUEUE EXPANDED

EZZ8763I

This message provides the IDS trace correlator and probe ID for the IDS event.

correlator is the IDS trace correlator associated with the event.

probe_id is the probe ID associated with the event.

EZZ8764I

This message provides the source IP address and source port from the IP packet associated with the IDS event. This message is issued only when the source IP address is relevant to the event.

source is the source IP address of the packet associated with the event.

port is the source IP port associated with the packet. The source port is only relevant for TCP or UDP protocols and will be zero for any other protocol. *port* will be zero if the source IP port is not known at the time of the attack. *port* will always be zero if *event_type* in message EZZ8762I is FAST SCAN DETECTED or SLOW SCAN DETECTED.

EZZ8765I

This message provides the destination IP address and destination port from the IP packet associated with the IDS event. This message is issued only when the destination IP address is relevant to the event.

dest is the destination IP address of the packet associated with the event.

port is the destination IP port associated wth the packet. The destination port is only relevant for TCP or UDP protocols and will be zero for any other protocol. *port* will be zero if the destination port is not known at the point that an attack event is detected.

EZZ8766I

This message provides the IDS policy rule name that is associated with the other messages in the group.

rule_name is the short name of the IDS rule that is associated with the messages in this group.

EZZ8767I

This message provides the IDS policy action name that is associated with the other messages in the group.

action_name is the short name of the IDS action that is associated with the messages in this group.

System Action: Processing Continues.

User or Operator Response: Save the system console log, IDS syslog file, and IDS packet trace for the person responsible for IDS policy definition. IDS policy definition determines if IDS events are written to syslog, the IDS packet trace, both or neither.

System Programmer Response: You can use the trmdstat OE shell command to analyze the IDS syslog file. You can use the IPCS trace formatters to format the IDS packet trace if one was collected for this event. If IDS policy is not maintained by the system programmer, then provide the log and trace information to the person responsible for IDS policy. You can use the IDS action name and IDS rule name to locate the IDS policy that is responsible for the messages.

If message EZZ8762I has an *event_type* of SCAN INTERVAL OVERRUN, scan processing is not able to complete an evaluation of the source ip addresses it is tracking in its normal internal interval (30 or 60 seconds). This might indicate that a large number of source ip addresses are being monitored. If the policy is using High scan sensitivity, consider lowering the scan sensitivity level for high usage ports. If message EZZ8762I has an *event_type* of SCAN STORAGE CONSTRAINED, determine the cause of the storage shortage. See the *z/OS Communications Server: IP Diagnosis* for information about storage shortages.

Source Data Set: EZBIDIDM

Procedure Name: EZBIDLOG

EZZ8762I IDS EVENT TYPE *event_type*

Explanation: This message is issued as part of a message group. See message EZZ8761I for a complete description of the message group.

EZZ8763I CORRELATOR *correlator* **- PROBEID** *probe_id*

Explanation: This message is issued as part of a message group. See message EZZ8761I for a complete description of the message group.

EZZ8764I SOURCE IP ADDRESS *source* **- PORT** *port*

Explanation: This message is issued as part of a message group. See message EZZ8761I for a complete description of the message group.

EZZ8765I DESTINATION IP ADDRESS *dest* **- PORT** *port*

Explanation: This message is issued as part of a message group. See messageEZZ8761I for a complete description of the message group.

EZZ8766I IDS RULE *rule_name*

Explanation: This message is issued as part of a message group. See message EZZ8761I for a complete description of the message group.

EZZ8767I IDS ACTION *action_name*

Explanation: This message is issued as part of a message group. See message EZZ8761I for a complete description of the message group.

EZZ8768I IDS SCAN STORAGE EXCEEDED nbrmeg MB, TRACKING nbrsip SOURCE IP ADDRESSES

Explanation: IDS scan-detection can consume large amounts of storage if HIGH was specified as the scan sensitivity for high usage ports or if the port is undergoing a flooding attack. If the storage used to track the source IP addresses exceeds 1 MB of storage, scan processing will inform the installation. This information can be used with message EZZ4364I to determine if scan detection is the cause of a storage constraint.

nbrmeg is the number of megabytes of storage exceeded. The message is issued at 1 MB, 2 MB and power of 2 MB increments (for example, 1 MB, 2 MB , 4 MB, 8 MB, and so forth).

nbrsip is the number of source IP addresses currently being tracked by scan detection.

System Action: Processing continues.

User or Operator Response: None

System Programmer Response: If the system is storage constrained and it appears that the scan detection is contributing to this problem, consider changing the scan policy. If the installation set the scan sensitivity to HIGH on high usage ports, consider reducing the sensitivity level or removing the port from scan detection until the storage constraint is resolved.

Source Data Set: EZBIDSCN

Procedure Name: EZBIDSCT

EZZ8769I ICMP WILL IGNORE REDIRECTS DUE TO INTRUSION DETECTION POLICY

Explanation: Intrusion Detection Policy is active and LIMIT was specified for the ICMP_REDIRECT Attack Policy. All further ICMP Redirects will be ignored.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBIDATK

Procedure Name: EZBIDAKP

EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR *TcpImage*

Explanation: The Policy Agent (Pagent) completed processing the configuration files and installing any active policies for the *TcpImage*. This message is displayed when policies have been processed for this *TcpImage* from Pagent configuration files during the following conditions:

- Policy Agent initialization.

- MODIFY procname,REFRESH.
- SIGHUP signal.
- Policy changes found during configuration file refresh interval processing.
- TCP/IP stack started.

TcpImage is the name of the TCP/IP stack for which the policies will be in effect.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PINIT, LDAPCLNT

Procedure Name: ReadLdapRules, pinit_fetch_policy_profile

EZZ8772I PAGENT LDAP POLICY PROCESSING COMPLETE FOR *TcpImage*

Explanation: The Policy Agent (Pagent) completed querying the LDAP server and installing any active policies for the *TcpImage*. This message is displayed when policies have been processed for this *TcpImage* from LDAP during the following conditions:

- Policy Agent initialization.
- MODIFY procname,REFRESH.
- SIGHUP signal.
- Policy changes found during LDAP server refresh interval processing.
- TCP/IP stack started.

TcpImage is the name of the TCP/IP stack for which the policies will be in effect.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LDAPCLNT

Procedure Name: ReadLdapRules

EZZ8773I PAGENT POLICIES PURGED FOR *TcpImage*

Explanation: The Policy Agent (Pagent) purged all policies for this *TcpImage*. This message is displayed in either of the following cases::

- If the PURGE option on the TCPIMAGE statement in Pagent configuration file was used, then this message is displayed when policies have been deleted for this *TcpImage* during Pagent shutdown.
- If a TCPIMAGE statement was deleted from the Pagent configuration file

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGENT, PINIT

Procedure Name: terminator, pinit_init_tcpimages

EZZ8774I PAGENT POLICIES NOT PURGED FOR *TcpImage*

Explanation: The policies for this *TcpImage* were not purged when the Policy Agent (Pagent) shut down. This message is displayed if the NOPURGE option is used on the TCPIMAGE statement in Pagent configuration file. This is the default option.

System Action: None.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PAGENT

Procedure Name: terminator

EZZ8791I *argument_description argument* **must be numeric**

Explanation: The **named** server is processing a command line numeric argument and found that it contains a value that is not numeric.

argument_description is the description of the argument being tested.

argument is the argument specified.

System Action: The **named** server ends.

User or Operator Response: Review the *z/OS Communications Server: IP Configuration Reference* for the correct options on the command line argument in error.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_int

EZZ8792I *argument_description argument* **out of range**

Explanation: The **named** server is processing a command line numeric argument and found that it did not convert to an integer properly because the command line numeric argument being processed is not in the range of 0 - 65535.

argument_description is the description of the argument being tested.

argument is the argument specified.

System Action: The **named** server ends.

User or Operator Response: Review the *z/OS Communications Server: IP Configuration Reference* for the correct options on the command line argument in error.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_int

EZZ8796I **port** *port_num* **out of range**

Explanation: The port number specified in the **named** server command line is not in the range of 1 - 65535.

port_num is the port number specified on the command line.

System Action: The **named** server ends.

User or Operator Response: Specify a port number in the range of 1 - 65535 and restart the **named** server.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ8797I **unknown option -***option*

Explanation: The user specified an incorrect option on the **named** server command line.

-*option* is the option that is considered to be incorrect.

System Action: The **named** server ends.

User or Operator Response: Correct or delete the incorrect command line option and restart the **named** server.

See the *z/OS Communications Server: IP Configuration Reference* for the correct command line options required to start the **named** server.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ8798I parsing options returned *command_line_option*

Explanation: An error occurred in the parsing of the **named** server command line.

command_line_option is the command line option specified.

System Action: The **named** server ends.

User or Operator Response: Correct or delete the incorrect command line option and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference* for the correct command line options required to start the **named** server.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ8799I extra command line arguments

Explanation: An error occurred in the parsing of the **named** server command line. There might be a duplicate command line argument or an argument specified without a value.

System Action: The **named** server ends.

User or Operator Response: See the *z/OS Communications Server: IP Configuration Reference* for the correct command line options required to start the **named** server.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ8800I ns_log_init() failed: *description*

Explanation: An error occurred while initializing the **named** server default logging channels possibly because of insufficient memory.

description describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO region address space or region size and restart the **named** server.

Source Data Set: NDMAIN

Procedure Name: main

EZZ8801I create_managers() failed: *description*

Explanation: An error occurred while initializing the **named** server thread managers.

description describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the

TC,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ8802I ns_omapi_init() failed: *description*

Explanation: An error occurred initializing the object managers.

description describes the error. attempt to start the object managers.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ8803I isc_app_start() failed: *description*

Explanation: An error occurred initializing the signal handlers for the **named** server.

description describes the error. attempt to start the signal handlers.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDMAIN

Procedure Name: main

EZZ8804I isc_mem_create() failed: *description*

Explanation: An error occurred allocating memory for the **named** server.

description describes the error. is

System Action: The **named** server stops loading.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart the **named** server. If the problem still occurs then re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDMAIN

Procedure Name: main

EZZ8807I fork(): *description* **- errno2=***errnojr*

Explanation: The **named** server could not create a child process because of insufficient resources to create another process, or the process already reached the maximum number of processes you can run, or the process requires more memory than is available.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that there are adequate resources available. Check the number of processes running and the time and either increase the limit or decrease the number running. Increase TSO address space or region size and restart the **named** server. If the problem still occurs then re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDOS

Procedure Name: ns_os_daemonize

EZZ8808I setsid(): *description*, **errno2=***errnojr*

Explanation: The **named** server could not create a session.

description is a message that describes the reason the new session could not be created.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the

Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDOS

Procedure Name: ns_os_daemonize

EZZ8809I chroot(): *description*, errno2=*errnojr*

Explanation: The **named** server could not change the root directory to the value specified by way of the **-t** command line variable most likely because the userid being used does not have superuser authority.

description is a message that describes the error from **named** servers attempt to set the root directory.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID starting the **named** server has superuser privileges. Verify that the root directory passed by the the command line actually exists and is a valid name.

Source Data Set: NDOS

Procedure Name: ns_os_daemonize

EZZ8810I chdir(/): *description*, errno2=*errnojr*

Explanation: The **named** server could not change the working directory to a root directory.

description is a message that describes the meaning of the attempt to set the working directory to root.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID starting the **named** server has superuser privileges.

Source Data Set: NDOS

Procedure Name: ns_os_daemonize

EZZ8811I user *user_id* unknown

Explanation: The username or ID number specified with the **-u** option on the command line does not exist.

user_id is an unknown z/OS UNIX System Services username or numeric user ID.

System Action: The **named** server ends.

User or Operator Response: Specify a valid z/OS UNIX System Services username or numeric user ID on the command line.

System Programmer Response: None.

Source Data Set: NDOS

Procedure Name: ns_os_inituserinfo

EZZ8812I initgroups(): *description*: errno2=*errnojr*

Explanation: The **named** is processing the userid passed from the **-u** command line option but determined that 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, or an MVS environmental or internal error occurred, or the System authorization facility (SAF) had an error, or **named** server is not authorized, only authorized users are allowed to alter the supplementary group IDs list.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Check the errnojr to determine if the maximum number of groups allowed was exceeded, or an invalid user is specified, or an MVS environmental or internal error occurred, or the System authorization facility (SAF) had an error, or **named** server is not authorized, only authorized users are allowed to alter the supplementary group IDs list. If you are not able to resolve this then re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: NDOS

Procedure Name: ns_os_inituserinfo

EZZ8814I setgid(): *description*: errno2=*errnojr*

Explanation: The attempt to set the group ID for the **named** server user ID failed.

description is a message that describes the error from setting the group ID.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: See the *z/OS Communications Server: IP Configuration Reference* and ensure that the **named** server is running with the correct privileges. Check with your security administrator to determine what security violation is being reported against the **named** server.

Source Data Set: NDOS

Procedure Name: ns_os_changeuser

EZZ8815I setuid(): *description*, errno2=*errnojr*

Explanation: The attempt to set the user ID for the **named** server failed.

description is a message that describes the error from setting the effective user id.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: See the *z/OS Communications Server: IP Configuration Reference* and ensure that the **named** server is running with the correct privileges. Check with your security administrator to determine what security violation is being reported against the **named** server.

Source Data Set: NDOS

Procedure Name: ns_os_changeuser

EZZ8816I couldn't malloc *description*: errno2=*errnojr*

Explanation: The attempt to allocate memory for the process ID file failed.

description is a message that describes the error from the attempt to allocate memory for the process ID file.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart **named** server.

Source Data Set: NDOS

Procedure Name: ns_os_writepidfile

EZZ8817I couldn't open pid file *description*: errno2=*errnojr*

Explanation: The **named** server failed to open the process ID file failed because of an error getting file information or the file being opened is not a z/OS UNIX regular file.

description is a message that describes the error from retrieving information about the process ID file.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the process ID file being used has a valid path and is a z/OS UNIX regular file and not a z/OS UNIX system file.

Source Data Set: NDOS

Procedure Name: ns_os_writepidfile

EZZ8818I could not fdopen() pid file *description*: errno2=*errnojr*

Explanation: The **named** failed while attempting to open the process ID file for writing.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the 4-byte Reason Code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server has write authority to the process id file.

Source Data Set: NDOS

Procedure Name: ns_os_writepidfile

EZZ8819I fprintf() to pid file *pid_filename* **failed**

Explanation: The **named** server failed to write to the process ID file.

pid_filename is the name of the process ID file.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The process ID file might be currupted. Delete the process ID file and restart the **named** server.

Source Data Set: NDOS

Procedure Name: ns_os_writepidfile

EZZ8820I fflush() to pid file *pid_filename* **failed**

Explanation: The **named** server attempted to write a buffer to the process ID file but found an end of file condition.

pid_filename is the name of the process ID file.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: The process ID file being used might be corrupt or the file space might be exhausted. Delete the process ID file or clean up the file system and restart the **named** server.

Source Data Set: NDOS

Procedure Name: ns_os_writepidfile

EZZ8821I undefined ACL *acl_name file_name*:*line_number*

Explanation: The **named** server referenced an access control list that was not defined.

acl_name is the name of an access control list.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the reference to the undefined access control listin the **named** server configuration file or define an access control list by that name.

System Programmer Response: None.

Source Data Set: ACLCONF

Procedure Name: convert_named_acl

EZZ8822I key name *key_name* **is not a valid domain name:** *file_name*:*line_number*

Explanation: The key name does not conform to the DNS naming standards.

key_name is the name of an security key.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the key name conforms to the naming standards described in RFC 1035. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC.

System Programmer Response: None.

Source Data Set: ACLCONF

Procedure Name: convert_keyname

EZZ8823I address match list contains unsupported element type: *file_name*:*line_number*

Explanation: An address match list in an access control list (ACL) contains an element that is not an IP address, key name or another ACL name.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that all ACL lists in the **named** server configuration file contain valid element types.

System Programmer Response: None.

Source Data Set: ACLCONF

Procedure Name: dns_acl_fromconfig

EZZ8824I cache cleaner could not create iterator: *description*

Explanation: There was not enough memory to initialize the **named** server cache cleaning process. The cache cleaning process will be set to an idle state and will be rescheduled.

description describes the error.

System Action: The **named** server continues. If a problem occurrs with the cache cleaner where another cache cleaner is still running then the **named** server will end with an assertion failure.

User or Operator Response: Notify Systems Programmer

System Programmer Response: Increase TSO address space or region size and restart the **named** server. If the **named** server does end with an assertion failure then re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator’s Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: DNSCACHE

Procedure Name: begin_cleaning

EZZ8825I master zones do not have an allow-notify field

Explanation: The allow-notify option is present in a master zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any master zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_set_allownotify

EZZ8826I stub zones do not have an allow-notify field

Explanation: The allow-notify option is present in a stub zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any stub zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_set_allownotify

EZZ8827I cache cleaner did not finish in one cleaning-interval

Explanation: The cache cleaner did not complete cleaning in the cleaning-interval. A cache cleaner is still active. The cleaning-interval is set in the options section of the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: If the cleaning-interval is set to a small value, increase it to allow more time for cache cleaning.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: cleaning_timer_action

EZZ8828I client *client_IP*: *description*

Explanation: An error occurred in a request from *client_IP* with a valid peer address.

client_IP is the IP address and port of the client peer address where the error described by *description* occurred.

description describes the error that occurred.

System Action: The **named** server continues.

User or Operator Response: Review the error message being reported for the peer socket. This message can occur for various error conditions from the **named** server client process. Set the debug level to 10 to produce debug messages from client processing.

System Programmer Response: None.

Source Data Set: CLIENT

Procedure Name: ns_client_logv

EZZ8829I client @*client_IP*: *description*

Explanation: An error occurred in a request from *client_IP* with an incorrect peer address.

client_IP is the incorrect IP address of the client in question where the error described by *description* occurred.

description describes the error that occurred.

System Action: The **named** server continues.

User or Operator Response: Review the error message being reported for the client IP address. This message can occur for various error conditions from the **named** server client process. Set the debug level to 10 to produce debug messages from client processing.

System Programmer Response: None.

Source Data Set: CLIENT

Procedure Name: ns_client_logv

EZZ8830I Out of memory

Explanation: The **named** server was not able to allocate space for an access control list table.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart the **named** server.

Source Data Set: CONFACL

Procedure Name: dns_c_acltable_new

EZZ8831I Failed to delete ACL element

Explanation: The **named** server is trying to delete access control lists from its internal access control list table and determined that There was an internal error detected while attempting to delete an access control list from its IP match list.

System Action: The **named** server ends.

User or Operator Response: This message should never be issued.

System Programmer Response: None.

Source Data Set: CONFACL

Procedure Name: dns_c_acltable_clear

EZZ8832I Not enough memory

Explanation: The **named** server was not able to allocate space for an access control list table.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart the **named** server.

Source Data Set: CONFACL

Procedure Name: dns_c_acl_new

EZZ8833I Not enough memory

Explanation: The **named** server was not able to allocate space for an access control list table.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart the **named** server.

Source Data Set: CONFACL

Procedure Name: dns_c_acl_new

EZZ8834I empty control statement

Explanation: The control statement in the **named** server configuration file is empty.

System Action: The **named** server ends.

User or Operator Response: Add valid entries to the control statement.

System Programmer Response: None.

Source Data Set: CONFCTL

Procedure Name: dns_c_ctrllist_validate

EZZ8835I type unix control channels are not implemented

Explanation: The control statement in the **named** server configuration file has a UNIX control channel. The UNIX control channel type from BIND 8 is not implemented in this version of the **named** server.

System Action: The **named** server continues.

User or Operator Response: Remove the UNIX control channel from the controls statement.

System Programmer Response: None.

Source Data Set: CONFCTL

Procedure Name: dns_c_ctrl_validate

EZZ8836I type inet control channel has no keys clause; control channel will be disabled

Explanation: The control statement in the **named** server configuration file has no keys clause.

System Action: The **named** server continues with no control channel configured.

User or Operator Response: Add a keys clause containing one or more valid keys.

System Programmer Response: None.

Source Data Set: CONFCTL

Procedure Name: dns_c_ctrl_validate

EZZ8837I type inet control channel has no keys; control channel will be disabled

Explanation: The control statement in the **named** server configuration file has no key in the keys clause.

System Action: The **named** server continues running with no control channel configured.

User or Operator Response: Add valid keys to the keys clause of the control statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTL

Procedure Name: dns_c_ctrl_validate

EZZ8838I option named-xfer is obsolete

Explanation: The option named-xfer is specified in the **named** server configuration file but is considered obsolete and is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove this option from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8839I hint zones do not have an allow-notify field

Explanation: The allow-notify option is present in a hint zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_set_allownotify

EZZ8840I forward zones do not have an allow-notify field

Explanation: The allow-notify option is present in a forward zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_set_allownotify

EZZ8841I option memstatistics-file is not implemented

Explanation: The option memstatistics-file is specified in the **named** server configuration file but is not implemented and is ignored.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Check the **named** server configuration file for the memstatistics-file option and remove it.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8842I the default for the auth-nxdomain option is now no

Explanation: The option auth-nxdomain was not found in the **named** server configuration file. The default setting changed from yes to no.

System Action: The **named** server will interpret option auth-nxdomain as having the value of no. Configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: If you need to specify auth-nxdomain as yes add it to your **named** server configuration file. See the *z/OS Communications Server: IP Configuration Reference* for the auth-nxdomain option necessary the meet your installations requirement.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8843I option deallocate-on-exit is obsolete

Explanation: The option deallocate-on-exit is obsolete and is ignored by the **named** server. **named** server always performs memory leak checking.

System Action: The **named** server will ignore this option. The configuration parser continues parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option deallocate-on-exit from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8844I master zones do not have an allow-notify field

Explanation: The allow-notify option is present in a master zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any master zones.

System Programmer Response: None.

Source Data Set: CONZONE

Procedure Name: dns_c_zone_getallownotify

EZZ8845I option fake-iquery is obsolete

Explanation: The option fake-iquery is obsolete and is ignored by the **named** server. This option was used in Bind 8 to enable simulating the obsolete DNS query type IQUERY. **named** server does not perform IQUERY simulation.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option fake-iquery from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8846I option fetch-glue is obsolete

Explanation: The option fetch-glue is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option fetch-glue from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8847I option has-old-clients is obsolete

Explanation: The option has-old-clients is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option has-old-clients from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8848I option host-statistics is not implemented

Explanation: The option host-statistics is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option host-statistics from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8849I option multiple-cnames is obsolete

Explanation: The option multiple-cnames is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option multiple-cnames from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8850I option rfc2308-type1 is not implemented

Explanation: The option rfc2308-type1 is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option rfc2308-type1 from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8851I option use-id-pool is obsolete

Explanation: The option use-id-pool is obsolete and is ignored by the **named** server. **named** server always allocates query IDs from a pool.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option use-id-pool from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8852I option treat-cr-as-space is obsolete

Explanation: The option treat-cr-as-space is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option treat-cr-as-space from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8853I option check-names is not implemented

Explanation: The option check-names is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8854I option use-ixfr is obsolete

Explanation: The option use-ixfr is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option use-ixfr from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8855I option max-cache-size is not implemented

Explanation: The option max-cache-size is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option max-cache-size from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8856I option min-roots is obsolete

Explanation: The option min-roots is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option min-roots from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8857I option serial-queries is obsolete

Explanation: The option serial-queries is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option serial-queries from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8858I option maintain-ixfr-base is obsolete

Explanation: The option maintain-ixfr-base is obsolete and is ignored by the **named** server. It was used in BIND 8 to determine whether a transaction log was kept for Incremental Zone Transfer. **named** server maintains a transaction log whenever possible. If you need to disable outgoing incremental zone transfers, use provide-ixfr no.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option maintain-ixfr-base from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8859I option max-ixfr-log-size is obsolete

Explanation: The option max-ixfr-log-size is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option max-ixfr-log-size from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8861I host portion of network address *address* is not zero

Explanation: The **named** server attempted to create an address match list but it found that the host portion of the network address being added to the address match list was not zero.

address is the network address specified.

System Action: The **named** server ends.

User or Operator Response: Verify that the network address being checked is a valid address.

System Programmer Response: None.

Source Data Set: CONFIP

Procedure Name: dns_c_imatchpattern_new

EZZ8862I view max-cache-size is not implemented

Explanation: The option max-cache-size, specified in the **named** configuration file under a view statment is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option max-cache-size from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_checkviews

EZZ8863I view check-names is obsolete

Explanation: The option check-names, specified in the **named** configuration file under a view statment is obsolete and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_checkviews

EZZ8864I stub zones do not have an allow-notify field

Explanation: The allow-notify option is present in a stub zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any stub zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallownotify

EZZ8865I option statistics-interval is not implemented

Explanation: The option statistics-interval is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option statistics-interval from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8866I option topology is deprecated

Explanation: The option topology is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option topology from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8867I hint zones do not have an allow-notify field

Explanation: The allow-notify option is present in a hint zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallownotify

EZZ8868I option rrset-order is not implemented

Explanation: The option rrset-order is not implemented and is ignored by the **named** server.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option rrset-order from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_checkconfig

EZZ8869I forward zones do not have an allow-notify field

Explanation: The allow-notify option is present in a forward zone in the **named** server configuration file. The option allow-notify is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-notify option from any forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallownotify

EZZ8870I bad transport value: *transport_type*

Explanation: The option check-names is obsolete and is ignored by the **named** server.

transport_type is the check-names setting from the options statement of the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_ctx_setchecknames

EZZ8871I bad transport value: *transport_type*

Explanation: The option check-names is obsolete and is ignored by the **named** server.

transport_type is the check-names setting from the options statement of the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_ctx_getchecknames

EZZ8872I bad transport value: *transport_type*

Explanation: The option check-names is obsolete and is ignored by the **named** server.

transport_type is the check_names setting from the options statement of the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFCTX

Procedure Name: dns_c_ctx_unsetchecknames

EZZ8873I dns_ipmatch_none element type

Explanation: The **named** server found no match for an address match list element type of **none** when deleting an element from the address match list table. The valid options for an address match list element are localhost, localnets, pattern, indirect, key, an access control list or any.

System Action: The **named** server might end.

User or Operator Response: Retry with a valid address match list element.

System Programmer Response: None.

Source Data Set: CONFIP

Procedure Name: dns_c_ipmatchelement_delete

EZZ8874I ipmatch none element type

Explanation: The **named** server found no match for an element type when copying an element from an address match list. The valid options for an address match list element are localhost, localnets, pattern, indirect, key, an access control list or any.

System Action: The **named** server will delete the element from its address match list or it will not be appended to its address match list.

User or Operator Response: Retry with a valid address match list element.

System Programmer Response: None.

Source Data Set: CONFIP

Procedure Name: dns_c_ipmatchelement_copy

EZZ8875I dns_ipmatch_none element type

Explanation: The **named** server found no match for an element type when printing an element from an address match list. The valid options for an address match list element are localhost, localnets, pattern, indirect, key, an access control list or any.

System Action: The **named** server will not print the address match list.

User or Operator Response: Retry with a valid address match list element.

System Programmer Response: None.

Source Data Set: CONFIP

Procedure Name: dns_c_ipmatchelement_print

EZZ8876I This type of channel doesn't have a path field

Explanation: The configuration file parser found a definition of a logging channel with an unnecessary file path option. The path option is only allowed for a logging channel definition containing a file option.

System Action: The **named** server continues parsing the **named** server configuration file.

User or Operator Response: Remove the logging channel file path option from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFLOG

Procedure Name: dns_c_logchan_setpath

EZZ8877I This type of channel doesn't have a version field

Explanation: The configuration file parser found a definition of a logging channel with an unnecessary version option. The versions option is only allowed for a logging channel definition containing a file option.

System Action: The **named** server continues parsing the **named** server configuration file.

User or Operator Response: Remove the logging channel versions option from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFLOG

Procedure Name: dns_c_logchan_setversions

EZZ8878I This type of channel doesn't have a size field

Explanation: The configuration file parser found a definition of a logging channel with an unnecessary size option specified. The size option is only allowed for a logging channel definition containing a file option.

System Action: The **named** server continues parsing the **named** server configuration file.

User or Operator Response: Remove the logging channel file size option from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFLOG

Procedure Name: dns_c_logchan_setsize

EZZ8879I This type of channel doesn't have a facility field

Explanation: The configuration file parser found a logging channel definition with an unnecessary facility option specified.

System Action: The **named** server continues parsing the **named** server configuration file.

User or Operator Response: Remove the logging channel facility option from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFLOG

Procedure Name: dns_c_logchan_setfacility

EZZ8880I Not a legal facility for a syslog channel: *facility*

Explanation: The configuration parser found a logging channel definition with a syslog option which specified an incorrect syslog_facility. Ensure that the syslog facility specified for the logging channel syslog is a valid syslog facility. If a channel goes to syslog, you can specify the facility to be any of the following:

- kern
- user
- mail
- daemon

- auth
- syslog
- lpr
- news
- uucp
- cron
- authpriv
- ftp
- local0
- local1
- local2
- local3
- local4
- local5
- local6
- local7

The default is daemon.

facility is the facility specified on the logging channel syslog option.

System Action: The **named** server continues parsing the **named** server configuration file.

User or Operator Response: Correct the problem by specifying a valid syslog facility.

System Programmer Response: None.

Source Data Set: CONFLOG

Procedure Name: dns_c_facility2string

EZZ8883I dns_c_parse_namedconf: error creating lexer

Explanation: There is not enough storage to startup the **rndc** utility.

System Action: **rndc** ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart **rndc**.

Source Data Set: CONFNDC

Procedure Name: parser_setup

EZZ8884I dns_c_parse_namedconf: error opening file *filename*

Explanation: The **rndc** utility is not able to open the **rndc** configuration file and lock it for read only. **named** server might need more memory to allow the **rndc** utility to process the **rndc** configuration file.

filename is the **rndc** configuration file that **rndc** is trying to open.

System Action: **rndc** ends.

User or Operator Response: Ensure that the file being opened is a valid file.

System Programmer Response: Increase TSO address space or region size and restart **rndc**.

Source Data Set: CONFNDC

Procedure Name: parser_setup

EZZ8885I dns_c_parse_namedconf: Error creating symtab

Explanation: There is not enough memory to create the **rndc** internal symbol table.

System Action: **rndc** ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart **rndc**.

Source Data Set: CONFNDC

Procedure Name: parser_setup

EZZ8886I dns_c_parse_namedconf: Error installing keyword

Explanation: The keyword might already exist in symbol table or there might not be enough memory to install the keyword in the internal symbol table.

System Action: **rndc.** configuration file parsing ends.

User or Operator Response: Review the **rndc** configuration file to determine if the keyword is already in use, else contact the system programmer.

System Programmer Response: Increase TSO addres space or region size and restart **rndc**.

Source Data Set: CONFNDC

Procedure Name: parser_setup

EZZ8887I *file_name*:*line_number error_text* **near** *token*

Explanation: The **rndc** configuation file parser found a problem in the **rndc** configuration file used to start the **rndc** utility.

file_name is the file name of the **rndc** configuration file.

line_number is the line number in the configuration file.

error_text is the error the parsing routine found.

token is the last token the parser accepted.

System Action: **rndc** configuration file parsing continues.

User or Operator Response: Correct **rndc** configuration file.

System Programmer Response: None.

Source Data Set: CONFNDC

Procedure Name: parser_complain

EZZ8888I *file_name*:*line_number error_text*

Explanation: The **rndc** configuation file parser found a problem in the **rndc** configuration file used to start the **rndc** utility.

file_name is the file name of the **rndc** configuration file.

line_number is the line number in the configuration file.

error_text is the error the parsing routine found.

System Action: **rndc** configuration file parsing continues.

User or Operator Response: Correct the **rndc** configuration file.

System Programmer Response: None.

Source Data Set: CONFNDC

Procedure Name: parser_complain

EZZ8889I Out of memory

Explanation: There is not enough memory to process view command in the **named** server configuration file.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and retry **rndc**.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_new

EZZ8890I zone *zone_name*: check-names is not implemented

Explanation: The option check-names is obsolete and is ignored by the **named** server.

zone_name is the name of the zone with the check-names option specified in the **named** server configuration file.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8891I view fetch-glue is obsolete

Explanation: The option fetch-glue, specified in the **named** server configuration file under a view statement, is obsolete and is ignored.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option fetch-glue from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_checkviews

EZZ8894I view rfc2308-type1 is not implemented

Explanation: The option rfc2308-type1, specified in the **named** configuration file under a view statement, is not implemented and is ignored.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** configuration file.

User or Operator Response: Remove option rfc2308-type1 from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_checkviews

EZZ8895I zone *zone_name*: pubkey is deprecated

Explanation: The option pubkey, specified in the **named** configuration file under a zone statement, is not implemented and is ignored. In BIND 8, this option was intended for specifying a public zone key for verification of signatures in DNSSEC signed zones when they are loaded from disk. **named** does not verify signatures on loading and ignores this option.

zone_name is the name of the zone with the pubkey option specified.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option pubkey from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8897I zone *zone_name***: maintain-ixfr-base is obsolete**

Explanation: The option maintain-ixfr-base, specified in the **named** configuration file under a zone statement, is not implemented and is ignored.

zone_name is the name of the zone with the maintain-ixfr-base option specified.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option maintain-ixfr-base from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8898I zone *zone_name***: ixfr-base is obsolete**

Explanation: The option ixfr-base, specified in the **named** configuration file under a zone statement, is not implemented and is ignored.

zone_name is the name of the zone with the ixfr-base option specified

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option ixfr-base from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8898I view min-roots is obsolete

Explanation: The option min-roots, specified in the **named** configuration file under a view statement, is obsolete and is ignored.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option min-roots from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_checkviews

EZZ8899I view rrset-order is not implemented

Explanation: The option rrset-order, specified in the **named** configuration file under a view statement, is not implemented and is ignored.

System Action: The **named** server will ignore this option. The configuration parser will continue parsing the remainder of the **named** server configuration file.

User or Operator Response: Remove option rrset-order from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_viewtable_checkviews

EZZ8900I Insufficient memory

Explanation: The **named** server ran out of system memory while processing a new view.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase TSO address space or region size and restart the **named** server.

Source Data Set: CONFVIEW

Procedure Name: dns_c_view_new

EZZ8901I bad transport value: *transport_type*

Explanation: The option check-names is obsolete and is ignored by the **named** server.

transport_type is the check-names setting from the view statement of the **named** server configuration file.

System Action: The **named** server ends.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_view_setchecknames

EZZ8902I bad transport value: *transport_type*

Explanation: The option check-names is obsolete and is ignored by the **named** server.

transport_type is the check-names setting from the view statement of the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_view_getchecknames

EZZ8903I bad transport value: *transport_type*

Explanation: The option check-names is obsolete and is ignored by the **named** server.

transport_type is the check-names setting from the view statement of the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: Remove option check-names from the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFVIEW

Procedure Name: dns_c_view_unsetchecknames

EZZ8904I zone *zone_name*: is disabled

Explanation: The **named** server configuration zone validation failed.

zone_name is the name of the zone found in error.

System Action: The **named** server continues.

User or Operator Response: Check affected zone definition for incorrect statements.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8905I zone *zone_name*: allow-update is ignored when update-policy is also used

Explanation: The allow-update option and the update-policy option cannot be defined together under a zone statement.

zone_name is the name of the zone found in error.

System Action: The **named** server continues.

User or Operator Response: Review the conflict between the allow-update option and the update-policy option for the affected zone in the **named** server configuration file and choose which one should remain active.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8906I zone *zone_name*: missing masters entry

Explanation: The masters option is missing under a slave zone statement in the **named** server configuration file. It is required to enable the slave zone *zone_name* to notify master zones for updates.

zone_name is the name of the zone found in error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Add the masters option under the affected slave zone statement.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8907I zone *zone_name*: masters value is empty

Explanation: The masters option under a slave zone statement in **named** server configuration file has no value.

zone_name is the name of the zone found in error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Specify at least one IP address of a master server on the masters option under the affected slave zone.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_validate

EZZ8908I hint zones do not have a pubkey field

Explanation: The option pubkey, specified in the **named** configuration file, is incorrect under a hint zone statement. The pubkey option is only allowed under master, slave or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the pubkey option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getpubkeylist

EZZ8909I forward zones do not have a pubkey field

Explanation: The option pubkey, specified in the **named** configuration file, is incorrect under a forward zone statement. The pubkey option is only allowed under master, slave or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the pubkey option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getpubkeylist

EZZ8910I forward zones do not have a file field

Explanation: The file option, specified in the **named** configuration file, is incorrect under a forward zone statement. The file option is only allowed under master, slave, stub or hint zones.

System Action: The **named** server ends.

User or Operator Response: Remove the file option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setfile

EZZ8911I forward zones do not have a file field

Explanation: The file option, specified in the **named** configuration file, is incorrect under a forward zone statement. The file option is only allowed under master, slave, stub or hint zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the file option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getfile

EZZ8912I hint zones do not have an allow-update-forwarding field

Explanation: The allow-update-forwarding option, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-update-forwarding option is only allowed under master, slave, stub or hint zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-update-forwarding option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowupdateforwarding

EZZ8913I forward zones do not have an allow-update-forwarding field

Explanation: The allow-update-forwarding option, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-update-forwarding option is only allowed under master, slave, stub or hint zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-update-forwarding option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowupdateforwarding

EZZ8914I hint zones do not have an allow-update-forwarding field

Explanation: The allow-update-forwarding option, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-update-forwarding option is only allowed under master, slave, stub or hint zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-update-forwarding option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowupdateforwarding

EZZ8915I forward zones do not have an allow-update-forwarding field

Explanation: The allow-update-forwarding option, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-update-forwarding option is only allowed under master, slave, stub or hint zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-update-forwarding option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowupdateforwarding

EZZ8916I slave zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a slave zone statement. The ssauth option is only allowed under master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ssuauth option from the slave zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setssuauth

EZZ8917I stub zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a stub zone statement. The ssauth option is only allowed under master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ssuauth option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setssuauth

EZZ8918I hint zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a hint zone statement. The ssauth option is only allowed under master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ssuauth option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setssuauth

EZZ8919I forward zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a forward zone statement. The ssauth option is only allowed under master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ssuauth option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setssuauth

EZZ8920I slave zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a slave zone statement. The ssauth option is only allowed under master zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the ssuauth option from the slave zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getssuauth

EZZ8921I stub zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a stub zone statement. The ssauth option is only allowed under master zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the ssuauth option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getssuauth

EZZ8922I hint zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a hint zone statement. The ssauth option is only allowed under master zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the ssuauth option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getssuauth

EZZ8923I forward zones do not have an ssuauth field

Explanation: The option ssuauth, specified in the **named** configuration file, is incorrect under a forward zone statement. The ssauth option is only allowed under master zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the ssuauth option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getssuauth

EZZ8924I hint zones do not have an allow-query field

Explanation: The option allow-query, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-query option is only allowed under master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the allow-query option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowquery

EZZ8925I forward zones do not have an allow-query field

Explanation: The option allow-query, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-query option is only allowed under master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the allow-query option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowquery

EZZ8926I hint zones do not have an allow-query field

Explanation: The option allow-query, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-query option is only allowed under master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the allow-query option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowquery

EZZ8927I forward zones do not have an allow-query field

Explanation: The option allow-query, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-query option is only allowed under master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the allow-query option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowquery

EZZ8928I hint zones do not have an allow-transfer field

Explanation: The option allow-transfer, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-transfer option is only allowed under master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-transfer option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowtransfer

EZZ8929I forward zones do not have an allow-transfer field

Explanation: The option allow-transfer, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-transfer option is only allowed under master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-transfer option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowtransfer

EZZ8930I hint zones do not have an allow-transfer field

Explanation: The option allow-transfer, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-transfer option is only allowed under master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the allow-transfer option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowtransfer

EZZ8931I forward zones do not have an allow-transfer field

Explanation: The option allow-transfer, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-transfer option is only allowed under master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the allow-transfer option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowtransfer

EZZ8932I hint zones do not have a dialup field

Explanation: The option dialup, specified in the **named** configuration file, is incorrect under a hint zone statement. The dialup option is only allowed under master, slave and stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the dialup option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setdialup

EZZ8933I forward zones do not have a dialup field

Explanation: The option dialup, specified in the **named** configuration file, is incorrect under a forward zone statement. The dialup option is only allowed under master, slave and stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the dialup option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setdialup

EZZ8934I hint zones do not have a dialup field

Explanation: The option dialup, specified in the **named** configuration file, is incorrect under a hint zone statement. The dialup option is only allowed under master, slave and stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the dialup option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getdialup

EZZ8935I forward zones do not have a dialup field

Explanation: The option dialup, specified in the **named** configuration file, is incorrect under a forward zone statement. The dialup option is only allowed under master, slave and stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the dialup option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getdialup

EZZ8936I stub zones do not have a notify field

Explanation: The option notify, specified in the **named** configuration file, is incorrect under a stub zone statement. The notify option is only allowed under master and slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotify

EZZ8937I hint zones do not have a notify field

Explanation: The option notify, specified in the **named** configuration file, is incorrect under a hint zone statement. The notify option is only allowed under master and slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotify

EZZ8938I forward zones do not have a notify field

Explanation: The option notify, specified in the **named** configuration file, is incorrect under a forward zone statement. The notify option is only allowed under master and slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotify

EZZ8939I stub zones do not have a notify field

Explanation: The option notify, specified in the **named** configuration file, is incorrect under a stub zone statement. The notify option is only allowed under master and slave zones.

System Action: The **named** server continues.

User or Operator Response: Remove the notify option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotify

EZZ8940I hint zones do not have a notify field

Explanation: The notify option is present in a hint zone in the **named** server configuration file. The notify option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the notify option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotify

EZZ8941I forward zones do not have a notify field

Explanation: The notify option is present in a forward zone in the **named** server configuration file. The notify option is only allowed in master zones or slave zones.

System Action: The **named** server continues. This option is ignored.

User or Operator Response: Remove the notify option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotify

EZZ8942I stub zones do not have a also_notify field

Explanation: The also-notify option is present in a stub zone in the **named** server configuration file. The also-notify option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the also-notify option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setalsonotify

EZZ8943I hint zones do not have a also_notify field

Explanation: The also-notify option is present in a hint zone in the **named** server configuration file. The also-notify option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the also-notify option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setalsonotify

EZZ8944I forward zones do not have a also_notify field

Explanation: The also-notify option is present in a forward zone in the **named** server configuration file. The also-notify option is only allowed in zones of type master or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the also-notify option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setalsonotify

EZZ8945I stub zones do not have a also_notify field

Explanation: The also-notify option is present in a stub zone in the **named** server configuration file. The also-notify option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the also-notify option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getalsonotify

EZZ8946I hint zones do not have a also_notify field

Explanation: The also-notify option is present in a hint zone in the **named** server configuration file. The also-notify option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the also-notify option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getalsonotify

EZZ8947I forward zones do not have a also_notify field

Explanation: The also-notify option is present in a forward zone in the **named** server configuration file. The also-notify option is only allowed in zones of type master or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the also-notify option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getalsonotify

EZZ8948I stub zones do not have a maintain-ixfr-base field

Explanation: The maintain-ixfr-base option is present in a stub zone in the **named** server configuration file. The maintain-ixfr-base option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the maintain-ixfr-base option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaintixfrbase

EZZ8949I hint zones do not have a maintain-ixfr-base field

Explanation: The maintain-ixfr-base option is present in a hint zone in the **named** server configuration file. The maintain-ixfr-base option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the maintain-ixfr-base option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaintixfrbase

EZZ8950I forward zones do not have a maintain-ixfr-base field

Explanation: The maintain-ixfr-base option is present in a forward zone in the **named** server configuration file. The maintain-ixfr-base option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the maintain-ixfr-base option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaintixfrbase

EZZ8951I stub zones do not have a maintain-ixfr-base field

Explanation: The maintain-ixfr-base option is present in a stub zone in the **named** server configuration file. The maintain-ixfr-base option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the maintain-ixfr-base option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaintixfrbase

EZZ8952I hint zones do not have a maintain-ixfr-base field

Explanation: The maintain-ixfr-base option is present in a hint zone in the **named** server configuration file. The maintain-ixfr-base option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the maintain-ixfr-base option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaintixfrbase

EZZ8953I forward zones do not have a maintain-ixfr-base field

Explanation: The maintain-ixfr-base option is present in a forward zone in the **named** server configuration file. The maintain-ixfr-base option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the maintain-ixfr-base option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaintixfrbase

EZZ8954I stub zones do not have an ixfr_base field

Explanation: The ixfr-base option is present in a stub zone in the **named** server configuration file. The option ixfr-base is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the ixfr_base option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setixfrbase

EZZ8955I hint zones do not have an ixfr_base field

Explanation: The ixfr-base option is present in a hint zone in the **named** server configuration file. The option ixfr-base is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the ixfr_base option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setixfrbase

EZZ8956I forward zones do not have an ixfr_base field

Explanation: The ixfr-base option is present in a forward zone in the **named** server configuration file. The option ixfr-base is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the ixfr_base option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setixfrbase

EZZ8957I stub zones do not have an ixfr_base field

Explanation: The ixfr-base option is present in a stub zone in the **named** server configuration file. The option ixfr-base is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ixfr_base option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getixfrbase

EZZ8958I hint zones do not have an ixfr_base field

Explanation: The ixfr-base option is present in a hint zone in the **named** server configuration file. The option ixfr-base is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ixfr_base option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getixfrbase

EZZ8959I forward zones do not have an ixfr_base field

Explanation: The ixfr_base option is specified in a forward zone in the **named** server configuration file. The ixfr_base option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ixfr_base option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getixfrbase

EZZ8960I stub zones do not have an ixfr-tmp-file field

Explanation: The option ixfr-tmp-file is specified in a stub zone in the **named** server configuration file. The ixfr-tmp-file option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the ixfr-tmp-file option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setixfrtmp

EZZ8961I hint zones do not have an ixfr-tmp-file field

Explanation: The option ixfr-tmp-file is specified in a hint zone in the **named** server configuration file. The ixfr-tmp-file option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the ixfr-tmp-file option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setixfrtmp

EZZ8962I forward zones do not have an ixfr-tmp-file field

Explanation: The ixfr-tmp-file option is specified in a forward zone in the **named** server configuration file. The ixfr-tmp-file option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the ixfr-tmp-file option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setixfrtmp

EZZ8963I stub zones do not have an ixfr-tmp-file field

Explanation: The option ixfr-tmp-file is specified in a stub zone in the **named** server configuration file. The ixfr-tmp-file option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ixfr-tmp-file option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getixfrtmp

EZZ8964I hint zones do not have an ixfr-tmp-file field

Explanation: The option ixfr-tmp-file is specified in a hint zone in the **named** server configuration file. The ixfr-tmp-file option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ixfr-tmp-file option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getixfrtmp

EZZ8965I forward zones do not have an ixfr-tmp-file field

Explanation: The option ixfr-tmp-file is specified in a forward zone in the **named** server configuration file. The ixfr-tmp-file option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the ixfr-tmp-file option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getixfrtmp

EZZ8966I hint zones do not have a pubkey field

Explanation: The pubkey option is specified in a hint zone in the **named** server configuration file. The pubkey option is only allowed in master zones ,slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the pubkey option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_addpubkey

EZZ8967I forward zones do not have a pubkey field

Explanation: The pubkey option is specified in a forward zone in the **named** server configuration file. The pubkey option is only allowed in master zones ,slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the pubkey option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_addpubkey

EZZ8968I master zones do not have a masters port field

Explanation: The masters port option is specified in a master zone in the **named** server configuration file. The masters port option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the master port option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmasterport

EZZ8969I hint zones do not have a masters port field

Explanation: The masters port option is specified in a hint zone in the **named** server configuration file. The masters port option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the master port option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmasterport

EZZ8970I forward zones do not have a masters port field

Explanation: The masters port option is specified in a forward zone in the **named** server configuration file. The masters port option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the master port option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmasterport

EZZ8971I master zones do not have a masters port field

Explanation: The masters port option is specified in a forward zone in the **named** server configuration file. The masters port option is only allowed in slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the master port option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmasterport

EZZ8972I hint zones do not have a masters port field

Explanation: The masters port option is specified in a hint zone in the **named** server configuration file. The masters port option is only allowed in slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the master-port option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmasterport

EZZ8973I forward zones do not have a masters port field

Explanation: The masters port option is specified in a forward zone in the **named** server configuration file. The masters port option is only allowed in slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the master port option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmasterport

EZZ8974I master zones do not have a masters IP addresses field

Explanation: The masters IP addresses option is specified in a master zone in the **named** server configuration file. The masters IP addresses option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the masters IP address option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmasterips

EZZ8975I hint zones do not have a masters IP addresses field

Explanation: The masters IP addresses option is specified in a hint zone in the **named** server configuration file. The masters IP addresses option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the masters IP address option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmasterips

EZZ8976I forward zones do not have a masters IP addresses field

Explanation: The masters IP addresses option is specified in a forward zone in the **named** server configuration file. The masters IP addresses option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the masters IP address option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmasterips

EZZ8977I master zones do not have a masters IP addresses field

Explanation: The masters IP addresses option is specified in a master zone in the **named** server configuration file. The masters IP addresses option is only allowed in slave zones, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the masters IP address option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmasterips

EZZ8978I hint zones do not have a masters IP addresses field

Explanation: The masters IP addresses option is specified in a hint zone in the **named** server configuration file. The masters IP addresses option is only allowed in slave zones, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the masters IP address option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmasterips

EZZ8979I forward zones do not have a masters IP addresses field

Explanation: The masters IP addresses option is specified in a forward zone in the **named** server configuration file. The masters IP addresses option is only allowed in slave zones, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the masters IP address option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmasterips

EZZ8980I forward zones do not have a zone-statistics field

Explanation: The statistics option is specified in a forward zone in the **named** server configuration file. The zone-statistics option is only allowed in master zones, slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the statistics option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setstatistics

EZZ8981I hint zones do not have a transfer-source field

Explanation: The transfer-source option is specified in a hint zone in the **named** server configuration file. The transfer-source option is only allowed in master zones, slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the transfer-source option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_settransfersource

EZZ8982I forward zones do not have a transfer-source field

Explanation: The transfer-source option is specified in a forward zone in the **named** server configuration file. The transfer-source option is only allowed in master zones, slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the transfer-source option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_settransfersource

EZZ8983I hint zones do not have a zone-statistics field

Explanation: The zone-statistics option is specified in a hint zone in the **named** server configuration file. The zone-statistics option is only allowed in master zones, slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the statistics option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setstatistics

EZZ8984I hint zones do not have a transfer-source field

Explanation: The transfer-source option is specified in a hint zone in the **named** server configuration file. The transfer-source option is only allowed in master zones, slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the transfer-source option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_gettransfersource

EZZ8985I forward zones do not have a transfer-source field

Explanation: The transfer-source option is specified in a forward zone in the **named** server configuration file. The transfer-source option is only allowed in master zones, slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the transfer-source option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_gettransfersource

EZZ8986I forward zones do not have a zone-statistics field

Explanation: The zone-statistics option is specified in a forward zone in the **named** server configuration file. The zone-statistics option is only allowed in master zones, slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the statistics option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getstatistics

EZZ8987I hint zones do not have a transfer-source-v6 field

Explanation: The transfer-source-v6 option is specified in a hint zone in the **named** server configuration file. The transfer-source-v6 option is only allowed in master zones, slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the transfer-source-v6 option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_settransfersourcev6

EZZ8988I forward zones do not have a transfer-source-v6 field

Explanation: The transfer-source-v6 option is specified in a forward zone in the **named** server configuration file. The transfer-source-v6 option is only allowed in master zones, slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the transfer-source-v6 option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_settransfersourcev6

EZZ8989I hint zones do not have a zone-statistics field

Explanation: The zone-statistics option is specified in a hint zone in the **named** server configuration file. The zone-statistics option is only allowed in master zones, slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the statistics option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getstatistics

EZZ8990I hint zones do not have a transfer-source-v6 field

Explanation: The option transfer-source-v6, specified in the **named** configuration file, is incorrect under a hint zone statement. The transfer-source-v6 option is only allowed in master zones, slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the transfer-source-v6 option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_gettransfersourcev6

EZZ8991I forward zones do not have a transfer-source-v6 field

Explanation: The option transfer-source-v6, specified in the **named** configuration file, is incorrect under a forward zone statement. The transfer-source-v6 option is only allowed in master zones, slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the transfer-source-v6 option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_gettransfersourcev6

EZZ8992I master zones do not have a max-transfer-time-in field

Explanation: The option max-transfer-time-in, specified in the **named** configuration file, is incorrect under a master zone statement. The max-transfer-time-in option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-time-in option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtranstimein

EZZ8993I hint zones do not have a max-transfer-time-in field

Explanation: The option max-transfer-time-in, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-time-in option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-time-in option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtranstimein

EZZ8994I forward zones do not have a max-transfer-time-in field

Explanation: The option max-transfer-time-in, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-time-in option is only allowed in slave zones, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-time-in option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtranstimein

EZZ8995I master zones do not have a max-transfer-time-in field

Explanation: The option max-transfer-time-in, specified in the **named** configuration file, is incorrect under a master zone statement. The max-transfer-time-in option is only allowed in slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-time-in option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtranstimein

EZZ8996I hint zones do not have a max-transfer-time-in field

Explanation: The option max-transfer-time-in, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-time-in option is only allowed in slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-time-in option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtranstimein

EZZ8997I forward zones do not have a max-transfer-time-in field

Explanation: The option max-transfer-time-in, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-time-in option is only allowed in slave zones, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-time-in option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtranstimein

EZZ8998I stub zones do not have a max-transfer-time-out field

Explanation: The option max-transfer-time-out, specified in the **named** configuration file, is incorrect under a stub zone statement. The max-transfer-time-out option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-time-out option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtranstimeout

EZZ8999I hint zones do not have a max-transfer-time-out field

Explanation: The option max-transfer-time-out, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-time-out option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-time-out option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtranstimeout

Chapter 9. EZZ9*xxxx* messages

EZZ9000I forward zones do not have a max-transfer-time-out field

Explanation: The option max-transfer-time-out, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-time-out option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-time-out option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtranstimeout

EZZ9001I stub zones do not have a max-transfer-time-out field

Explanation: The option max-transfer-time-out, specified in the **named** configuration file, is incorrect under a stub zone statement. The max-transfer-time-out option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-time-out option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtranstimeout

EZZ9002I hint zones do not have a max-transfer-time-out field

Explanation: The option max-transfer-time-out, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-time-out option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-time-out option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtranstimeout

EZZ9003I forward zones do not have a max-transfer-time-out field

Explanation: The option max-transfer-time-out, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-time-out option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-time-out option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtranstimeout

© Copyright IBM Corp. 1994, 2002

EZZ9004I master zones do not have a max-transfer-idle-in field

Explanation: The option max-transfer-idle-in, specified in the **named** configuration file, is incorrect under a master zone statement. The max-transfer-idle-in option is only allowed in slave zones or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-idle-in option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtransidlein

EZZ9005I hint zones do not have a max-transfer-idle-in field

Explanation: The option max-transfer-idle-in, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-idle-in option is only allowed in slave zones or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-idle-in option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtransidlein

EZZ9006I forward zones do not have a max-transfer-idle-in field

Explanation: The option max-transfer-idle-in, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-idle-in option is only allowed in slave zones or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-idle-in option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtransidlein

EZZ9007I master zones do not have a max-transfer-idle-in field

Explanation: The option max-transfer-idle-in, specified in the **named** configuration file, is incorrect under a master zone statement. The max-transfer-idle-in option is only allowed in slave zones or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-idle-in option from the master zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtransidlein

EZZ9008I hint zones do not have a max-transfer-idle-in field

Explanation: The option max-transfer-idle-in, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-idle-in option is only allowed in slave zones or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-idle-in option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtransidlein

EZZ9009I forward zones do not have a max-transfer-idle-in field

Explanation: The option max-transfer-idle-in, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-idle-in option is only allowed in slave zones or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-idle-in option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtransidlein

EZZ9010I stub zones do not have a max-transfer-idle-out field

Explanation: The option max-transfer-idle-out, specified in the **named** configuration file, is incorrect under a stub zone statement. The max-transfer-idle-out option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-idle-out option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtransidleout

EZZ9011I hint zones do not have a max-transfer-idle-out field

Explanation: The option max-transfer-idle-out, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-idle-out option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-idle-out option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtransidleout

EZZ9012I forward zones do not have a max-transfer-idle-out field

Explanation: The option max-transfer-idle-out, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-idle-out option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-transfer-idle-out option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxtransidleout

EZZ9013I stub zones do not have a max-transfer-idle-out field

Explanation: The option max-transfer-idle-out, specified in the **named** configuration file, is incorrect under a stub zone statement. The max-transfer-idle-out option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-idle-out option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtransidleout

EZZ9014I hint zones do not have a max-transfer-idle-out field

Explanation: The option max-transfer-idle-out, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-transfer-idle-out option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-idle-out option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtransidleout

EZZ9015I forward zones do not have a max-transfer-idle-out field

Explanation: The option max-transfer-idle-out, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-transfer-idle-out option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-transfer-idle-out option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxtransidleout

EZZ9016I slave zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a slave zone statement. The sig-validity-interval option is only allowed in master zones.

System Action: The **named** server ends.

User or Operator Response: Remove the sig-validity-interval option from the slave zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setsigvalidityinterval

EZZ9017I stub zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a stub zone statement. The sig-validity-interval option is only allowed in master zones.

System Action: The **named** server ends.

User or Operator Response: Remove the sig-validity-interval option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setsigvalidityinterval

EZZ9018I hint zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a hint zone statement. The sig-validity-interval option is only allowed in master zones.

System Action: The **named** server ends.

User or Operator Response: Remove the sig-validity-interval option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setsigvalidityinterval

EZZ9019I forward zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a forward zone statement. The sig-validity-interval option is only allowed in master zones.

System Action: The **named** server ends.

User or Operator Response: Remove the sig-validity-interval option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setsigvalidityinterval

EZZ9020I slave zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a slave zone statement. The sig-validity-interval option is only allowed in master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the sig-validity-interval option from the slave zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getsigvalidityinterval

EZZ9021I stub zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a stub zone statement. The sig-validity-interval option is only allowed in master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the sig-validity-interval option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getsigvalidityinterval

EZZ9022I hint zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a hint zone statement. The sig-validity-interval option is only allowed in master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the sig-validity-interval option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getsigvalidityinterval

EZZ9023I forward zones do not have a sig-validity-interval field

Explanation: The option sig-validity-interval, specified in the **named** configuration file, is incorrect under a forward zone statement. The sig-validity-interval option is only allowed in master zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the sig-validity-interval option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getsigvalidityinterval

EZZ9024I stub zones do not have a max-ixfr-log-size field

Explanation: The option max-ixfr-log-size, specified in the **named** configuration file, is incorrect under a stub zone statement. The max-ixfr-log-size option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-ixfr-log-size option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxixfrlog

EZZ9025I hint zones do not have a max-ixfr-log-size field

Explanation: The option max-ixfr-log-size, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-ixfr-log-size option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-ixfr-log-size option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxixfrlog

EZZ9026I forward zones do not have a max-ixfr-log-size field

Explanation: The option max-ixfr-log-size, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-ixfr-log-size option is only allowed in master zones or slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-ixfr-log-size option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxixfrlog

EZZ9027I stub zones do not have a max-ixfr-log-size field

Explanation: The option max-ixfr-log-size, specified in the **named** configuration file, is incorrect under a stub zone statement. The max-ixfr-log-size option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-ixfr-log-size option from the stub zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxixfrlog

EZZ9028I hint zones do not have a max-ixfr-log-size field

Explanation: The option max-ixfr-log-size, specified in the **named** configuration file, is incorrect under a hint zone statement. The max-ixfr-log-size option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-ixfr-log-size option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxixfrlog

EZZ9029I forward zones do not have a max-ixfr-log-size field

Explanation: The option max-ixfr-log-size, specified in the **named** configuration file, is incorrect under a forward zone statement. The max-ixfr-log-size option is only allowed in master zones or slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the max-ixfr-log-size option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxixfrlog

EZZ9030I hint zones do not have a forward field

Explanation: The option forward, specified in the **named** configuration file, is incorrect under a hint zone statement. The forward option is only allowed in master, slave, stub or forward zones.

System Action: The **named** server ends.

User or Operator Response: Remove the forward option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setforward

EZZ9031I hint zones do not have a forward field

Explanation: The option forward, specified in the **named** configuration file, is incorrect under a hint zone statement. The forward option is only allowed in master, slave, stub or forward zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the forward option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getforward

EZZ9032I hint zones do not have a forwarders field

Explanation: The option forwarders, specified in the **named** configuration file, is incorrect under a hint zone statement. The forwarders option is only allowed in master, slave, stub or forward zones.

System Action: The **named** server ends.

User or Operator Response: Remove the forwarders option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setforwarders

EZZ9033I hint zones do not have a forwarders field

Explanation: The option forwarders, specified in the **named** configuration file, is incorrect under a hint zone statement. The forwarders option is only allowed in master, slave, stub or forward zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the forwarders option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getforwarders

EZZ9034I hint zones do not have an allow-update field

Explanation: The option allow-update, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-update option is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-update option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowupd

EZZ9035I forward zones do not have an allow-update field

Explanation: The option allow-update, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-update option is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the allow-update option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setallowupd

EZZ9036I hint zones do not have an allow-update field

Explanation: The option allow-update, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-update option is only allowed in master, slave, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the allow-update option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowupd

EZZ9037I forward zones do not have an allow-update field

Explanation: The option allow-update, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-update option is only allowed in master, slave, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the allow-update option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getallowupd

EZZ9038I hint zones do not have an allow-update field

Explanation: The option allow-update, specified in the **named** configuration file, is incorrect under a hint zone statement. The allow-update option is only allowed in master, slave, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the allow-update option from the hint zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_unsetallowupd

EZZ9039I forward zones do not have an allow-update field

Explanation: The option allow-update, specified in the **named** configuration file, is incorrect under a forward zone statement. The allow-update option is only allowed in master, slave, or stub zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the allow-update option from the forward zone statement in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_unsetallowupd

EZZ9040I dispatch *thread_id*: *error_text*

Explanation: The **named** server detected an error on a code thread being dispatched for execution. The **named** server will report the error and return to determine next action.

thread_id is the hexadecimal address of the thread in error.

error_text describes the error.

System Action: The **named** server might end.

User or Operator Response: Set the debug message level no higher than 92 in the **named** configuration file to determine if there are any other dispatching errors. If you are unable to fix the error then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 92. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- Dump the **named** server address space if the server continues running after this message was issued. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping address spaces.

Source Data Set: DISPATCH

Procedure Name: dispatch_log

EZZ9041I ns_clientmgr_create() failed: *description*

Explanation: The **named** server failed to create an interface client manager because of insufficient memory.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_create

EZZ9042I creating TCP socket: *description*

Explanation: The **named** server could not open a TCP socket for the interface manager because of one of the following reasons:

- Insufficient memory
- Not able to place a mutually exclusive lock
- An incorrect TCP family was specified
- A socket allocation failure occurred

description describes the error.

System Action: The **named** server continues.

User or Operator Response: If the problem is insufficient memory, then contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart **named** server. If the problem still occurs then re-create this problem with a SYSTCPIP component trace active specifying the **CSOCKET** option and set the debug level on the **named** log files to three. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_accepttcp

EZZ9043I binding TCP socket: *description*

Explanation: The **named** server could not bind the new interface manager TCP socket because of the following reasons:

- There is a mismatch on the socket address family.
- Permission is denied.
- The address specified is not valid on this host.
- The address is already in use.
- The socket is already bound to an address.

The **named** server detaches the interface manager socket being created.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **CSOCKET** option and set the debug level on the **named** log files to three. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_accepttcp

EZZ9044I listening on TCP socket: *description*

Explanation: The **named** server failed to listen on the bound interface manager TCP socket because of the following:

- The socket parameter is not a valid socket descriptor.
- The socket is not bound to a local address, and the protocol does not support listening on an unbound socket.
- An invalid argument was supplied. The socket is not named (a bind() was not done), or the socket is ready to accept connections (a listen() was already done). The socket is already connected.
- Insufficient system resources are available to complete the call.
- The descriptor is for a file, not for a socket.
- The socket parameter is not a socket descriptor that supports the listen() call.

The **named** server detaches the interface manager socket being created.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **CSOCKET** option and set the debug level on the **named** log files to three. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_accepttcp

EZZ9045I no longer listening on *IP_addr#port_num*

Explanation: The **named** server stopped listening for queries on the interface *IP_addr* on port *port_num*. The generation number is not current. The **named** server might be shutting down or cleaning up interfaces that have gone away or changed addresses.

IP_addr is the IP address.

port_num is the port number.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: purge_old_interfaces

EZZ9046I listening on IPv4 interface *interface_name*, *IP_addr#port_num*

Explanation: The **named** server is listening on the IPv4 interface at *IP_addr* on port *port_num*.

interface_name is the name of the interface the **named** server did not find a match for in the **listen-on** options statement in the **named** server configuration file.

IP_addr is the IP address.

port_num is the port number.

System Action: The **named** server continues.

User or Operator Response: Review the **listen-on** option in the **named** configuration file and determine if the

address in the match list is the correct address for the **named** server to listen on.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: do_ipv4

EZZ9047I creating IPv4 interface *interface_name* **failed; interface ignored**

Explanation: The **named** server could not create a link to the *interface_name* interface because of one of the following:

- Insufficient memory.
- Unable to lock the memory context for the interface manager because of insufficient memory or incorrect security privileges.

interface_name is the IPv4 interface the **named** server is creating.

System Action: The **named** server ignores this interface. The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: do_ipv4

EZZ9048I ignoring IPv4 interface *interface_name IP_addr#port_num*

Explanation: The **named** server could not create a link to the *interface_name* interface. This could be because of an incorrect IPv4 address or because of insufficient memory to resize the access control list (ACL).

interface_name is the IPv4 interface name.

IP_addr is the IP address.

port_num is the port number.

System Action: The **named** server continues.

User or Operator Response: Ensure that the IP adress is the correct format. Contact the system programmer to increase memory.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: do_ipv4

EZZ9049I bad IPv6 listen-on list: must be any or none

Explanation: The **named** server does not bind a separate socket to each IPv6 interface address as with IPv4. As a result, the configuration file option listen-on-v6 can only have the access control list **any** or **none** as its parameter.

System Action: The **named** server continues.

User or Operator Response: Review the **named** configuration file and correct the incorrect listen-on-v6 address match list argument.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: do_ipv6

EZZ9050I listening on IPv6 interfaces, port *port_num*

Explanation: The **named** server is listening on all IPv6 interfaces on port *port_num*.

port_num is the port number the **named** server will use to listen for any IPv6 incoming queries.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: do_ipv6

EZZ9051I listening on IPv6 interfaces failed

Explanation: The **named** server could not create a link to the IPv6 interfaces possibly because of insufficient memory.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: do_ipv6

EZZ9052I no IPv6 interfaces found

Explanation: The **named** server determined the system could not support IPv6 interfaces.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: ns_interfacemgr_scan

EZZ9053I no IPv4 interfaces found

Explanation: The **named** server determined the system could not support IPv4 interfaces.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: ns_interfacemgr_scan

EZZ9054I not listening on any interfaces

Explanation: The **named** server did not bind to any IPv4 or IPv6 interfaces. Since the **named** server is not listening on any interfaces, it has no way to respond to any queries.

System Action: The **named** server continues.

User or Operator Response: Ensure that the **named** server configuration file correctly identifies the interface to be listened on for your environment.

System Programmer Response: None.

Source Data Set: INTRFCEM

Procedure Name: ns_interfacemgr_scan

EZZ9055I dialup option *dialup_option* **cannot be used with master zones**

Explanation: The valid dialup options for master zones are **no**, **yes** and **notify**.

dialup_option is the dialup option specified in the **named** configuration file under a zone statement.

System Action: The **named** server ends.

User or Operator Response: Remove the dialup option in the master zone or change the option to **no**, **yes** or **notify** as required by your configuration requirements. See the *z/OS Communications Server: IP Configuration Reference* for information about the Bind 9 DNS dialup option.

System Programmer Response: none.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setdialup

EZZ9056I dialup option *dialup_option* **cannot be used with stub zones**

Explanation: The valid dialup options for stub zones are **passive** and **notifypassive**.

dialup_option is the dialup option specified in the **named** configuration file under a zone statement.

System Action: The **named** server ends.

User or Operator Response: Remove the dialup option in the stub zone or change the option to **passive** or **notifypassive** as required by your configuration requirements. See the *z/OS Communications Server: IP Configuration Reference* for information about the Bind 9 DNS dialup option.

System Programmer Response: none.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setdialup

EZZ9057I stub zones do not have a notify-source field

Explanation: The Ensure that there-source option is present in a stub zone in the **named** server configuration file. The option notify-source is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify-source option from any stub zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotifysource

EZZ9058I exiting the named server abnormally, BIND v9

Explanation: The **named** server is shutting down because of an error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the TC, UDP, and IOCTL options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: NDMAIN

Procedure Name: cleanup

EZZ9059I *file_name*: **seek:** *description*

Explanation: The **named** server failed to complete a seek on the journal. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_seek

EZZ9060I *file_name*: **read:** *description*

Explanation: An error other than end-of-file occurred while reading a journal record from the *file_name* journal. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_read

EZZ9061I *file_name*: **write:** *description*

Explanation: The **named** server failed to write a journal record to the *file_name* journal. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_write

EZZ9062I *file_name*: **flush:** *description*

Explanation: The **named** server failed to flush the journal write buffer during journal file syncronization. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_fsync

EZZ9063I *file_name*: **fsync:** *description*

Explanation: The **named** server failed to write the journal data from the file buffer to the storage device. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_fsync

EZZ9064I *file_name*: **create:** *description*

Explanation: The **named** server failed to create the journal file. The error described by *description* will explain the errno from the fopen(). The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_file_create

EZZ9065I *file_name*: **write:** *description*

Explanation: The **named** server failed to write to the journal file. The journal file name will match the zone being journaled. The error described by *description* will explain the errno from the fwrite(). The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_file_create

EZZ9066I *file_name*: **close:** *description*

Explanation: The **named** server failed to close the journal file. The journal file name will match the zone being journaled. The error described by *description* will explain the errno from the fclose(). The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_file_create

EZZ9067I **journal file** *file_name* **does not exist, creating it**

Explanation: The **named** server did not find a journal file associated with a zone for the dynamic update or incremental file transfer, IXFR. When the **named** server does not find a journal file in such a case, it creates one automatically with the file name zone_file_name.jnl where zone_file_name is the name of the zone being journaled.

file_name is the journal file name.

System Action: The **named** server creates the journal file.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_open

EZZ9068I *file_name*: **open:** *description*

Explanation: The journal file being opened exists but the **named** server could not open it. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

description describes the error.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_open

EZZ9069I *file_name*: **journal format not recognized**

Explanation: The **named** server determined the journal file contains an incorrect header. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_open

EZZ9070I *file_name*: **journal unexpectedly empty**

Explanation: The **named** server is reading a journal file but found it empty. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_open

EZZ9071I *file_name*: **journal file corrupt: expected serial** *journal_serial_no*, **got** *transaction_serial_no*

Explanation: There was an internal inconsistency between the journal serial number and the transaction serial number. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

journal_serial_no is the serial number for a valid journal transaction.

transaction_serial_no is the serial number for the journal transaction header being processed.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_next

EZZ9072I *file_name*: **offset too large**

Explanation: The **named** server failed to process a journal header because it was too large for the space available and would cause a wrap condition. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: journal_next

EZZ9073I malformed transaction: *number* **SOAs**

Explanation: The basic journal transaction consistancy check failed. There were more than two SOAs (start of authority) resource records involved in the changes being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

number is the number of SOA resource records found.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Ensure that there is only one SOA record defined to the zone being journaled.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_commit

EZZ9074I malformed transaction: serial number would decrease

Explanation: The **named** server rejected a transaction that did not properly increment the zone serial number. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

System Action: The **named** server continues without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_commit

EZZ9075I malformed transaction: *file_name* **last serial** *journal_serial* **!= transaction first serial** *trans_serial*

Explanation: The **named** server rejected a transaction whose zone serial number was incorrectly incrememented. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

journal_serial is journal header ending serial number.

trans_serial is the transaction serial number.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_commit

EZZ9076I *file_name***: journal file corrupt: missing initial SOA**

Explanation: The journal file does not contain an initial start of authority (SOA) resource record. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: roll_forward

EZZ9077I journal open failure

Explanation: The **named** server failed to open a journal to print its contents possibly because of insufficient memory. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Ensure that the **named** server has the authority to open the journal.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: JOURNAL

Procedure Name: dns_journal_print

EZZ9078I *file_name*: **journal file corrupt: missing initial SOA**

Explanation: The journal file does not contain an initial start of authority (SOA) resource record. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name: is the journal file name.

System Action: The **named** server continues.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_print

EZZ9079I *file_name*: **cannot print: journal file corrupt**

Explanation: The **named** server failed to print the journal file because it is corrupt. The journal file name will match the zone being journaled.

file_name is the journal file name.

System Action: Journal is not printed.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_journal_print

EZZ9080I journal corrupt: empty transaction

Explanation: The journal file had a transaction with an empty header. The journal file might be corrupt.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: read_one_rr

EZZ9081I *file_name*: **journal file corrupt: expected serial** *journal_serial*, **got** *trans_serial*

Explanation: The **named** server is advancing to the next journal entry but found a journal entry where a zone start of authority (SOA) serial field is not the same. Whenever a zone is dynamically updated and the update is stored in the zones journal file, the journal entry includes a copy of the zones SOA serial field before and after the update. The journal file name will match the zone being journaled.

file_name is the journal file name.

journal_serial is the current zones SOA serial field.

trans_serial is the zone SOA serial field of the journal transaction read.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: read_one_rr

EZZ9082I *file_name*: **journal corrupt: impossible RR size (***RR_size* **bytes)**

Explanation: The journal had a resource record (RR) size of less than 11 or greater than 65,535 bytes. The journal file might be corrupt. The journal file name will match the zone being journaled. The incremental file transfer (IXFR) might fail or an authoritative master file might be incomplete.

file_name is the journal file name.

RR_size is the size of the incorrect resource record.

System Action: The **named** server continues running without processing the journal file.

User or Operator Response: Delete the corrupt journal file and restart the **named** server to create a new journal file.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: read_one_rr

EZZ9083I **unknown logging category** *category_name* **ignored**

Explanation: An incorrect logging category was specified in the **named** server configuration file. The valid category names can be found in the *z/OS Communications Server: IP Configuration Reference*.

category_name is the incorrect logging category specified.

System Action: The **named** server continues.

User or Operator Response: See the *z/OS Communications Server: IP Configuration Reference* for information about coding logging channel categories in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: LOGCONF

Procedure Name: category_fromconf

EZZ9085I logging channel *channel_name*: *description*

Explanation: The **named** server failed to assign a logging channel to a logging context for the logging statements in the **named** configuration file. Messages logged to the failing logging channel will not be available.

channel_name is the channel name assigned to the logging context.

description describes the error.

System Action: The **named** server ends.

User or Operator Response: Ensure that the file name assigned to the logging catalog specified in the **named** configuration file matches a logging channel. See the *z/OS Communications Server: IP Configuration Reference* for information about coding logging channels in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: LOGCONF

Procedure Name: category_fromconf

EZZ9086I file log channel has no file name

Explanation: A file log channel defined in the **named** server configuration file has no file name assigned to it. The log files path might not be correctly assigned to the logging channel because of insufficient memory.

System Action: The **named** server continues.

User or Operator Response: The **named** server should have caught this error when processing the **named** configuration file as this error is caught by the configuration parser. Contact the system programmer to increase memory.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: LOGCONF

Procedure Name: channel_fromconf

EZZ9087I hint zones do not have a notify-source field

Explanation: The notify-source option is present in a hint zone in the **named** server configuration file. The option notify-source is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify-source option from any hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotifysource

EZZ9088I forward zones do not have a notify-source field

Explanation: The notify-source option is present in a forward zone in the **named** server configuration file. The option notify-source is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify-source option from any forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotifysource

EZZ9089I exiting named, BIND v9 (due to early fatal error)

Explanation: A fatal error occurred while the **named** server was initializing.

System Action: The **named** server ends.

User or Operator Response: Review the previously logged error messages in the SYSLOG DAEMON logging file to determine the cause of the fatal error.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: ns_main_earlyfatal

EZZ9090I *source_file*:*line_num*: *assertion_type*(*condition*) **failed**

Explanation: This is a fatal error. The **named** server failed because of an internal requirement not being met.

source_file is the source data set name were the assertion failure occurred.

line_num is the line number in the source data set where the assertion failure occurred.

assertion_type is the type of assertion that failed.

condition is the condition that failed.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: NDMAIN

Procedure Name: assertion_failed

EZZ9091I exiting named, BIND v9 (due to assertion failure)

Explanation: A fatal failure occurred.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: NDMAIN

Procedure Name: assertion_failed

EZZ9092I *file_name*:*line_no*: **fatal error:**

Explanation: A fatal error occurred in one of the base **named** server library routines.

file_name is the name of the file where the library error occurred.

line_no is the line number in the file where the library error occurred.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: NDMAIN

Procedure Name: library_fatal_error

EZZ9093I exiting named, BIND v9 (due to fatal error in library)

Explanation: This is a fatal error. A catastrophic error occurred in one of the base **named** server library routines.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: NDMAIN

Procedure Name: library_fatal_error

EZZ9094I *source_file*:*line_no*: **unexpected error:**

Explanation: There was an unexpected error in one of the base **named** server library functions.

file_name is the name of the file where the library error occurred.

line_no is the line number in the file where the library error occurred.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: NDMAIN

Procedure Name: library_unexpected_error

EZZ9095I starting named, BIND *version_no*

Explanation: The **named** server starts the load procedure.

version_no is the version of the BIND 9 DNS **named** server being started.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ9096I exiting named, BIND v9

Explanation: The **named** server is shutting down.

System Action: The **named** server ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: cleanup

EZZ9097I dumping master file: *tempfile_name***: open:** *description*

Explanation: The **named** server failed to create or open a unique file.

tempfile_name is the name of the unique temporary file being created and opened.

description describes the error.

System Action: The **named** server stops the dump procedure and continues.

User or Operator Response: Ensure that there is adequate space to create the temporary file.

System Programmer Response: None.

Source Data Set: MASTERDU

Procedure Name: dns_master_dump

EZZ9098I dumping master file: *tempfile_name***:** *description*

Explanation: There was an error dumping an entire zone database into a temporary master file possibly because of insufficient memory. The **named** server deletes the temporary dump file.

tempfile_name is the name of the unique temporary file being used to contain the entire zone database being dumped.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: This problem might be resolved by increasing the TSO address space or region size and restarting the **named** server. Otherwise, re-create this problem with a SYSTCPIP component trace active specifying the **TC**,**UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.

- Use **rndc dumpdb** to dump the **named** servers cache to a file. See the *z/OS Communications Server: IP System Administrator's Commands* on the use of the **rndc** utility.
- Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: MASTERDU

Procedure Name: dns_master_dump

EZZ9099I dumping master file: *tempfile_name*: **close:** *description*

Explanation: The **named** server could not close the temporary dump output file.

tempfile_name is the name of the unique temporary file being closed.

description describes the error.

System Action: The **named** server deletes the temporary dump file and continues.

User or Operator Response: Ensure that there is adequate space on the output device to contain the entire zone database being dumped.

System Programmer Response: None.

Source Data Set: MASTERDU

Procedure Name: dns_master_dump

EZZ9100I dumping node to file: *file_name*: **open:** *description*

Explanation: The **named** server could not open the dump output file.

file_name is the name of the file being opened as output.

description describes the error.

System Action: The **named** server does not complete the dump of a node to a master file.

User or Operator Response: Review the file that the **named** server is attempting to open to make sure it exists and that the **named** server has write access to it.

System Programmer Response: None.

Source Data Set: MASTERDU

Procedure Name: dns_master_dumpnode

EZZ9101I dumping master file: rename: *file_name*: *description*

Explanation: The **named** server could not rename the temporary dump file to the specified dump *file_name*. The **named** server does not delete the dump file.

file_name is the name the unique temporary file is being renamed.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Review the error description and fix the problem. Rename the file name manually if required.

System Programmer Response: None.

Source Data Set: MASTERDU

Procedure Name: dns_master_dump

EZZ9102I stub zones do not have a notify-source field

Explanation: The notify-source option is present in a stub zone in the **named** server configuration file. The option notify-source is only allowed in slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the notify-source option from any stub zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotifysource

EZZ9103I hint zones do not have a notify-source field

Explanation: The notify-source option is present in a hint zone in the **named** server configuration file. The option notify-source is only allowed in slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the notify-source option from any hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotifysource

EZZ9104I unimplemented channel command *omapi_region*

Explanation: The **named** server could not process the **rndc** reload zones or reload configuration because of an unimplemented function. The valid **rndc** commands are as follows:

- reload
- reload zone
- refresh zone
- stats
- querylog
- dumpdb
- stop
- halt

The **named** server informs **rndc** that the channel command is not implemented.

omapi_region is the object manager name.

System Action: The **named** server continues.

User or Operator Response: Contact the user that issued the **rndc** command to inform him of the valid **rndc** commands.

System Programmer Response: None.

Source Data Set: OMAPI

Procedure Name: control_setvalue

EZZ9105I unknown channel command *omapi_region*

Explanation: The **named** server determined that the **rndc** channel command is incorrect. The valid **rndc** commands are as follows:

- reload
- reload zone
- refresh zone

- stats
- querylog
- dumpdb
- stop
- halt

omapi_region is the object manager name.

System Action: The **named** server continues.

User or Operator Response: Contact the user that issued the **rndc** command to inform him of the valid **rndc** commands.

System Programmer Response: None.

Source Data Set: OMAPI

Procedure Name: control_setvalue

EZZ9106I couldn't find key *key_id* **for use with command channel** *socket_addr*: *file_name*:*line_number*

Explanation: The **named** server failed to find the key *key_id* defined in a key statement. The control facility will be unavailable for use with the key *key_id*

key_id is the id of the key being searched for in the key id list.

socket_addr is the socket IP address for the object manager being configured.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Ensure that the key *key-id* is defined in the **named** configuration file.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: register_keys

EZZ9107I unsupported algorithm *alg_name* **in key** *key_id* **for use with command channel** *socket_addr*: *file_name*:*line_number*

Explanation: The **named** server control facility only supports keys generated with the hmac-md5 algorithm. The control facility will be unavailable for use with the key *key_id*

alg_name is the name of the algorithm requested to use to interpret the key registration.

key_id is the id of the key being registered.

socket_addr is the socket IP address for the object manager being configured.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Ensure that the key *key_id* is using algorithm hmac-md5. See the *z/OS Communications Server: IP Configuration Reference* for instructions about supplying an algorithm with a key in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: register_keys

EZZ9108I can't use secret for key *key_id* **on command channel** *socket_addr*: *description*: *file_name*:*line_number*

Explanation: The **named** server failed to decode the key *key_id*. The key specified might have been created incorrectly. The control facility will be unavailable for use with the key *key_id*

key_id is the name of the key being decoded.

socket_addr is the socket IP address for the object manager being configured.

description describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Ensure that the key is correctly generated. See the *z/OS Communications Server: IP System Administrator's Commands* for information about using the **dnssec-keygen** utility to generating keys.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: register_keys

EZZ9109I couldn't register key *key_id* **for use with command channel** *socket_addr*: *description*

Explanation: There was a problem registering the *key_id* because of insufficient memory or the key being registered already exists. The control facility will be unavailable for use with the key *key_id*.

key_id is the name of the key being registered.

socket_addr is the socket IP address for the object manager being configured.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the key being registered is unique for the object manager being authorized. Contact the system programmer to increase memory.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: OMAPCONF

Procedure Name: register_keys

EZZ9110I couldn't install new ACL for command channel *socket_addr*: *description*: *file_name*:*line_number*

Explanation: The **named** server is updating the access control list (ACL) with the new key information but found there is already a listener for this socket address or the **named** server failed to create the new access control list possibly because of insufficient memory. The control facility will be unavailable for use on the address *socket_addr*. The old access list will remain.

socket_addr is the socket IP address for the object manager listener being updated.

description describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: If the message is issued because there is already a listener for this socket, and you are relying on automatic rndc configuration, you should stop the **named** server, remove any existing **control** statements from named.conf, remove /etc/rndc.conf if it exists, and restart the **named** server. If you persistently have storage problems with the **named** server after previously allocating storage, or you are not restricting the REGION for

the application, there might be a storage leak. In that case, provide a storage dump of the **named** server address space and contact the IBM Support Center.

Source Data Set: CNTLCONF

Procedure Name: update_listener

EZZ9111I command channel listening on *socket_addr*

Explanation: The **named** server control facility is operational on *socket_addr*.

socket_addr is the socket IP address for the object manager listener being started.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: add_listener

EZZ9112I couldn't add command channel *socket_addr*: *description*: *file_name*:*line_number*

Explanation: The **named** server failed to establish the control facility listener for the address *socket_addr*. This might be because of insufficient memory or not being able to create a lock. The control facility will be unavailable for use on the address *socket_addr*.

socket_addr is the socket IP address for the object manager listener.

description describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the region size and restart the **named** server. Ensure that the user ID used by the **named** server the correct security privileges.

Source Data Set: CNTLCONF

Procedure Name: add_listener

EZZ9113I no key statements for use by control channels

Explanation: The **named** server found no key list information. The control facility will be unavailable for use on the address *socket_addr*.

System Action: The **named** server continues.

User or Operator Response: Ensure that the **named** server configuration file contains a controls option with the keys option specified with a valid key list.

System Programmer Response: None.

Source Data Set: OMAPCONF

Procedure Name: ns_omapi_configure

EZZ9115I *operation*: *description*

Explanation: The *operation* specified failed because of *description* The **named** server will clean up any allocated resources. This message is a general error handling message.

operation is the operation that failed.

description is the reason the *operation* failed.

System Action: The **named** server continues.

User or Operator Response: Resolve the error based on the description reported.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: CHECKM macro

EZZ9116I could not get query source dispatcher

Explanation: The **named** server failed to get a dispatcher for the resolver of a view because of insufficient memory or a socket could not be created.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the PORTRANGE statement in the TCPIP.PROFILE data set does not reserved a high number of ports. Otherwise, increase the TSO address space or region size and restart the **named** server.

Source Data Set: SERVER

Procedure Name: get_view_querysource_dispatch

EZZ9119I no root hints for view *view_name*

Explanation: The view *view_name*, specified in the **named** server configuration file, lacks a hint zone of class Internet, or IN. If multiple views are specified, each one must contain a hint zone. This view is not configured.

view_name is the view which is specifying the hint zone in error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the hint zone specified under the *view_name* view statement has a resource record, RR, class of IN.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_view

EZZ9120I zone *zone_name***: wrong class for view** *view_name*

Explanation: The *zone_name* class does not match the view *view_name* class name.

zone_name is the zone name under the view statement in the **named** server file.

view_name is the view name in the **named** server file.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the zone class specified under a view matches the view class.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_zone

EZZ9121I forward zones do not have a notify-source field

Explanation: The option notify-source is specified on a forward zone statement in the **named** server configuration file. The option notify-source is only allowed for slave zones. This option is ignored.

System Action: The **named** server continues.

User or Operator Response: Remove the notify-source option from the forward zone statment in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotifysource

EZZ9122I zone *zone_name*: file not specified

Explanation: The **named** server found that hint zone *zone_name* does not have a file statement.

zone_name is the name of the hint zone.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Specify a file name to be used by the hint zone *zone_name*

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_zone

EZZ9123I ignoring non-root hint zone *zone_name*

Explanation: The hint zone *zone_name* specified is not a root zone. The **named** server ignores the zone.

zone_name is the name of the hint zone.

System Action: The **named** server continues.

User or Operator Response: Ensure that the hint zone *zone_name* is a root zone.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_zone

EZZ9124I stub zones do not have a notify-source-v6 field

Explanation: The option notify-source-v6 is specified on a stub zone statement in the **named** server configuration file. The option notify-source-v6 is only allowed for slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify-source-v6 option from any stub zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotifysourcev6

EZZ9125I change directory to *directory_path* failed: *description*

Explanation: The **named** server could not change the working directory to the path specified by the directory option in the **named** server configuration file.

directory_path is the directory path name.

description describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the specified path exists and that the **named** server user ID has read/write access to the path.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: options_callback

EZZ9126I loading configuration from *conf_file*

Explanation: The **named** server is loading its configuration from the **named** server configuration file *conf_file*. This file was specified on the **named** server command line option **-c**

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: load_configuration

EZZ9127I when using view statements, all zones must be in views

Explanation: The **named** server found a zone specified in the **named** server configuration file that was not in the scope of a view.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure all zones belong to views.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: load_configuration

EZZ9128I ignoring config file logging statement due to -g option

Explanation: The **named** server bypasses the logging configuration specified in the **named** server configuration file because the **-g named** command line option is specified. The **-g** option causes the **named** server to execute as a foreground process and all message will be sent to stderr. The z/OS UNIX shell does not support sending stderr to the screen. You must redirect stderr to a file.

System Action: The **named** server continues.

User or Operator Response: If you do not want the **named** server to run as a foreground process then shut down and restart the **named** server without specifying the **-g** command line option.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: load_configuration

EZZ9130I named, BIND *version* **is running**

Explanation: The **named** server starts processing.

version is the version of DNS Bind9 that is currently active.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: run_server

EZZ9131I shutting down *flushing_changes_indicator*

Explanation: The **named** server is shutting down. If the **named** server detects that it has changes to flush, then the *flushing_changes_indicator* variable will be specified as **: flushing changes**

flushing_changes_indicator will indicate if changes are being flushed, otherwise, this field is blank.

System Action: The **named** server ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: shutdown_server

EZZ9132I *action***:** *description*

Explanation: A catastrophic error occurred in the **named** server operation.

action is the action the **named** server is performing.

description describes the error.

System Action: The **named** server ends.

User or Operator Response: Review any previously logged messages in the **named** server log files or the SYSLOG DAEMON logging file and correct the problem reported.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: fatal

EZZ9133I exiting named, BIND v9 (due to fatal error)

Explanation: A catastrophic error occurred in the **named** server operation.

System Action: The **named** server ends.

User or Operator Response: This message is expected when TCP/IP ends. In any other case, review any previously logged messages in the **named** server log files or the SYSLOG DAEMON logging file and correct the problem reported.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: fatal

EZZ9134I reloading configuration failed: *description*

Explanation: An error occurred while the **named** server was reloading its **named** server configuration file.

description describes the error.

System Action: The **named** server stops reloading and continues.

User or Operator Response: Review any previously logged error messages in the **named** server log files and correct the problem reported. Retry the **named** reload.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: ns_server_reload

EZZ9135I reloading zones failed: *description*

Explanation: An error occurred while the **named** server was reloading the zone statements from its **named** server configuration file.

description describes the error.

System Action: The **named** server stops reloading and continues.

User or Operator Response: Correct any previously logged error messages in the **named** server log files and retry the **named** reload.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: ns_server_reload

EZZ9136I dynamic update failed: *reason* **(***error_text***), updating zone** *zone/class*

Explanation: The **named** server failed a dynamic update. The cause of the error is described by the *reason* and the accompanying errno. The update requested will be denied.

reason is the reason the dynamic update failed.

error_text is the errno in text accompanying the error.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Correct the error being reported and retry the update.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: FAILC macro

EZZ9137I dynamic update error: *reason***:** *errno_text***, updating zone** *zone/class*

Explanation: A dynamic update request to the **named** server failed because of a problem opening or writing to a journal file possibly because of insufficient memory. The dynamic update request is not processed.

reason is the reason the dynamic update failed.

errno_text is the errno in text describing the error.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Set the logging channel debug level to no more than eight to generate dynamic update debug messages. Correct the problem to enable journaling during dynamic updates. If the problem was caused by insufficient memory, increase the region size.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: FAILS macro

EZZ9138I could not get zone keys for secure dynamic update

Explanation: The **named** server attempted to generate new signature (SIG) and next, NXT records to go along with dynamically updated information but it could not find the necessary zone keys. The update requested is not processed.

System Action: The **named** server continues.

User or Operator Response: Ensure that the appropriate zone key files (zone.key and zone.private) are present in the working directory.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_signatures

EZZ9139I could not create update response message: *description*

Explanation: The **named** server is saving the query transaction signature (TSIG) status, because the query was signed, but failed to reserve buffer space in the reply for the TSIG. The update request is not processed.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: UPDATE

Procedure Name: respond

EZZ9140I update RR has incorrect class *update_class***, updating zone** *zone*/*class*

Explanation: The **named** server received a dynamic update request in which the client, **nsupdate**, attempted to update a resource record (RR) with an incorrect type where the class is not **ANY.**. The client which issued the dynamic update might not be compatible with the IBM **nsupdate** utility. The update request is denied.

update_class is the class in error.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Specify dynamic update request with a valid non-meta (not ANY, AXFR, MAILA, or MAILB) record type. See RFC2136 on details of the dynamic update protocol. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9141I attempt to add CNAME alongside non-CNAME ignored, updating zone *zone*/*class*

Explanation: The **named** server failed to add a canonical name (CNAME) resource record because the name of the CNAME record to be added already exists as another CNAME, next (NXT) or signature (SIG) resource record. The resource record update being processed is ignored and the dynamic update request continues.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Ensure that the name of the CNAME resource record being added is unique.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9142I attempt to add non-CNAME alongside CNAME ignored, updating zone *zone/class*

Explanation: The **named** server failed to add a resource record because a canonical name (CNAME) resource record exists with the same name.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Either change the name of the CNAME record in the zone database or change the name of the resource record being added and try the update request again.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9143I attempt to create 2nd SOA ignored, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update but found that the update requested will create another start of authority (SOA) record which is not allowed. The resource record database can only have one SOA record. The resource record update being processed is ignored and the dynamic update request continues.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Remove the SOA resource record from the update.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9144I SOA update failed to increment serial, ignoring it, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update but found that the start of authority (SOA) record being processed incorrectly caused the SOA serial number to decrease or stay unchanged relative to the SOA in the data base. The resource record update being processed is ignored and the dynamic update request continues.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: When using the **nsupdate** utility to increment the SOA serial number ensure that the new serial number is greater than the zone SOA serial number. See RFC2315 section 3.4.2.2 for further discussion. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9145I adding an RR, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and is adding a resource record (RR).

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9147I delete all rrsets from a name, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and is preparing to delete associated resource record sets. A check will be make to prohibit the deletion of the start of authority (SOA) or name server (NS) records where the name is equal to the zone name. The resource record sets (rrsets) will be deleted by the **named** server dynamic update.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9148I attempt to delete all SOA or NS records ignored, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and reports that an attempt was made to delete all resource records that were of type start of authority (SOA) or name server (NS).

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues. The record is not deleted and the dynamic update is ignored by named, and the named server continues processing any additional update requests.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9149I deleting an rrset, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and is deleting resource record set (rrset) records that are not of type start of authority (SOA) or name server (NS).

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9150I attempt to delete SOA ignored, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and determined that the record being deleted is a start of authority (SOA) record. SOA records cannot be deleted.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Ensure that there is no dynamic update request to delete the SOA record. The dynamic update request might originate from the use of the **nsupdate** utility or from a request from a **DHCP** server.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9151I attempt to delete last NS ignored, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and determined that one name server (NS) record exists. The last NS record is not deleted.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Ensure that there is no dynamic update request to delete the last NS record. The dynamic update request might originate from the use of the **nsupdate** utility or from a request from a **DHCP** server.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9152I deleting an RR, updating zone *zone/class*

Explanation: The **named** server is performing a dynamic update and is deleting resource records that have identical resource data. Unique resource records will not be deleted.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9153I SIG/NXT update failed: *error_text*, **updating zone** *zone/class*

Explanation: The **named** server is performing a dynamic update and is updating signature (SIG) or next (NXT) records in a secured database but discovered that the zone might not have any keys. The dynamic update ends.

error_text describes the error.

zone is the zone that was being updated at the time of the error.

class is the class of the zone, which is usually **IN**.

System Action: The **named** server continues.

User or Operator Response: Ensure that the secured database being updated has valid keys defined in its zone data.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9154I **validating** *dns_name region_used*: *error_text*

Explanation: The validator encountered an error described by *error_text*.

dns_name is the DNS name.

region_used is the name of the buffer region.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Set the debug level to at least three to generate more dispatcher debug messages. Correct the error being reported.

System Programmer Response: None.

Source Data Set: VALIDATO

Procedure Name: validator_logv

EZZ9155I **validator @***validator_pointer*: *error_text*

Explanation: The validator encountered an error described by *error_text*.

validator_pointer is the address of the validator.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Set the debug level to at least three to generate more dispatcher debug messages. Correct the error being reported.

System Programmer Response: None.

Source Data Set: VALIDATO

Procedure Name: validator_logv

EZZ9156I **transfer of** *zone_name* **from** *master_zone*: *msg_text*

Explanation: This message is used to log the incoming zone transfer messages during the zone transfer in process.

zone_name is the zone name being transfered.

master_zone is the master zone is being transferred from.

msg_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Correct the error for the zone being transferred. Set the debug level to three but no higher than seven to produce zone transfer debug messages.

Source Data Set: XFRIN

Procedure Name: xfrin_logv

EZZ9157I *request_type* **request denied:** *error_text*

Explanation: Either an incremental zone transfer (IXFR) or authoritative zone transfer (AXFR) quota was exceeded. The transfer will fail. The maximum number of outbound zone transfers that can be run concurrently is set in the **named** server configuration file as a **transfers-out** option. The default is 10.

request_type is the type of request and can be either of the following:

- IXFR for an incremental zone transfer.
- AXFR for an authoritative zone transfer.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Limit the number of zone transfers being processed to the number specified by the **transfers-out** option or increase the value specified by the **transfers-out** option.

System Programmer Response: None.

Source Data Set: XFROUT

Procedure Name: ns_xfr_start

EZZ9158I **hint zones do not have a notify-source-v6 field**

Explanation: The notify-source-v6 option is specified in a hint zone in the **named** server configuration file. The option notify-source-v6 is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify-source-v6 option from any hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotifysourcev6

EZZ9159I **IBM message catalog** *msgcat_name* **open failed** *description*

Explanation: The **named** server failed to open its message catalog. Default internal English messages will be used.

msgcat_name is the name of the message catalog that failed to open.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure a valid message catalog file named ns9.cat exists in the path specified by the NLSPATH z/OS UNIX shell environment variable.

System Programmer Response: None.

Source Data Set: B9MSGCAT

Procedure Name:

EZZ9161I **IBM message catalog** *msgcat_name* **closed**

Explanation: The **named** server closed its message catalog as part of its shut down process.

msgcat_name is the name of the message catalog.

System Action: The **named** server continues to shut down.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: B9MSGCAT

Procedure Name:

EZZ9162I Ignoring DNS_VERSION environment variable value of *environment_variable*

Explanation: The **named** server found the environment variable DNS_VERSION set to the value of **tso**. The value of **tso** is not longer supported. The correct values for the DNS_VERSION environment variable should be set to one of the following:

- v9 for BIND 9 DNS
- V4 for BIND 4 DNS

System Action: The **named** server continues.

User or Operator Response: Ensure that the DNS_VERSION environment variable is set to **v9** for BIND v9 or **v4** for BIND v4.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ9163I Found DNS_VERSION environment variable with value of *environment_variable*

Explanation: The **named** server found the environment variable DNS_VERSION defined in the z/OS UNIX shell and is using its value to determine what version of BIND to activate.

environment_variable is the value assigned to the DNS_VERSION environment variable.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ9164I -n value of *-n_value* **out of range, value of** *max_cpus* **being used**

Explanation: The **named** server determined that the **-n** command line variable value specified for the number of CPUs the **named** server is allocated exceeds the maximum. The **named** server will use the maximum number of CPUs instead of what was specified by the *-n* command line variable.

-n_value is the number of cpu's specified from the **-n** command line variable.

max_cpus is the maximum number of cpus.

System Action: The **named** server continues.

User or Operator Response: Determine the number of CPUs required to support the **named** for your environment and specify that number for the **-n** command line variable.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ9165I Defaulting to BIND version 4

Explanation: The **named** server attempted to determine which version of BIND to start from either the environment variable DNS_VERSION or the **-V** command line argument. Because neither was found, BIND v4 was activated.

System Action: The BIND v9 **named** server will activate the BIND v4 **named** server and the BIND 9 **named** server ends.

User or Operator Response: If you do not want BIND v4 then set either the DNS_VERSION environment to **v9** or the **-V** command line argument to **v9** and restart the **named** server.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: version_check

EZZ9166I starting named, BIND v4

Explanation: The BIND v4 version of the **named** server is being started.

System Action: The BIND v9 **named** server ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: version_check

EZZ9167I Call of BIND4 failed, errno=*errno*, **errnojr=***errnojr*

Explanation: The BIND 9 **named** server issued the command required to start BIND v4 **named** server but failed because of one of the following reasons:

- There are insufficient resources to create another process, or else the process already reached the maximum number of processes you can run.
- The process requires more space than is available.

The BIND v4 **named** server did not start.

errno is the decimal Return Code. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Review the errno and errnojr codes to determine if the system has enough memory available and if another process can be created. Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server Ensure that the number of processes active on the system does not exceed the configured maximum.

Source Data Set: NDMAIN

Procedure Name: version_check

EZZ9168I Unable to dynamically determine number of CPUs

Explanation: The **named** server is unable to determine how many logical CPUs are available for use when running in a stand-alone environment and not in an LPAR or under VM. The number of CPUs is set to the default value.

System Action: The **named** server continues.

User or Operator Response: If **named** server needs a specific number of CPUs, then use the **-n** command line variable to specify that amount.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9169I Using default of *number_cpus* **CPUs for -n option**

Explanation: The **named** server is unable to determine how many CPUs are available.

number_cpus is the default number of CPUs used.

System Action: The **named** server continue.

User or Operator Response: If **named** server needs a specific number of CPUs, then use the **-n** command line variable to specify that amount.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9170I *config_level* **mode detected. Using** *number_CPUs_configured* **CPUs for -n option**

Explanation: The **named** server is unable to determine the total number of CPUs that are in the configuration.

config_level is the configuration level

number_CPUs_configured is the total number of CPUs that are in the configuration.

System Action: The **named** server continues.

User or Operator Response: If **named** server needs a specific number of CPUs, then use the -n command line variable to specify that amount.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9171I *config_level* **mode detected. Using** *number_CPUs_configured* **CPUs for -n option**

Explanation: The **named** server is unable to determine the total number of CPUs that are in the configuration.

config_level is the configuration level

number_CPUs_configured is the total number of CPUs that are in the configuration.

System Action: The **named** server continues.

User or Operator Response: If **named** server needs a specific number of CPUs, then use the -n command line variable to specify that amount.

System Programmer Response:

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9172I *config_level* **mode detected. Using** *number_CPUs_configured* **CPUs for -n option**

Explanation: The **named** server is unable to determine the total number of CPUs that are in the configuration.

config_level is the configuration level

number_CPUs_configured is the total number of CPUs that are in the configuration.

System Action: The **named** server continues.

User or Operator Response: If **named** server needs a specific number of CPUs, then use the -n command line variable to specify that amount.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9173I not started with Superuser authority. Exiting named, BIND v9

Explanation: The **named** server must be started by a user with the Superuser privilege.

System Action: The **named** server ends.

User or Operator Response: Obtain Superuser status for the user ID being used to start the **named** server from the RACF administrator. See the *z/OS Communications Server: IP Configuration Reference* for information about superuser status.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: check_user_authorization

EZZ9174I *function*: **zone** *zone_name*: *error_message*

Explanation: The **named** server uses this message to log zone processing errors.

function is the name of the function reporting the error.

zone_name is the name of the zone being processed.

error_message describes the error.

System Action: The **named** server continues.

User or Operator Response: Correct the error described in the *error_message*.

System Programmer Response: None.

Source Data Set: ZONE

Procedure Name: zone_log

EZZ9175I *source_file*:*line_number*: **file does not end with newline**

Explanation: An unexpected end-of-file condition caused the **named** server to end the parsing of the master file.

source_file is the name of the master file being processed.

line_number is the line number in the master file being processed.

System Action: The **named** server continues.

User or Operator Response: Ensure that the master file data being processed is not a partial file.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: WARNUNEXPECTEDEOF

EZZ9176I dns_master_load: *source_file*:*line_number*: **unexpected end of** *type*

Explanation: An unexpected end-of-line or end-of-file condition caused the **named** server to end the parsing of the master file.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

type is the token type being processed.

System Action: The **named** server continues.

User or Operator Response: Review the master file data being processed for possible undesired charcters or partial files.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: gettoken

EZZ9177I dns_master_load: *source_file*:*line_number*: **$TTL** *ttl_value* > **MAXTTL, setting $TTL to 0**

Explanation: The **named** server is processing a time-to-live (TTL) value and found the value to be greater than 2147483647 seconds. The **named** server will set the TTL value to zero and continue to process the master file.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

ttl_value value found for the TTL being processed.

System Action: The **named** server continues.

User or Operator Response: Review the record being processed in the master file and correct its TTL value.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9178I dns_master_load: *source_file*:*line_number*: **$INCLUDE may not be used with $DATE**

Explanation: The **named** server was loading data from a master file and encountered a $INCLUDE directive after previously processing a $DATE directive. $INCLUDE should not be used after $DATE is used in the same file. The zone containing the error will not load.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

System Action: The **named** server continues.

User or Operator Response: Remove the $DATE directive from *source_file* at line *line_number*.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9179I dns_master_load: *source_file*:*line_number*: **unknown $ directive** *directive*

Explanation: The **named** server was processing master file records and found a directive that is in error. Master file processing ends. The valid directives are as follows:

- $ORIGIN
- $INCLUDE
- $TTL
- $GENERATE

source_file is the name of the master file being processed.

line_number is the line number in the master file.

directive is the directive being processed that is in error.

System Action: The **named** server continues.

User or Operator Response: Remove the unknown directive from the master file.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9180I dns_master_load: *source_file*:*line_number*: **No current owner name**

Explanation: The **named** server is processing a master file and found an error condition which indicates that the owner name in this master file is not found. The owner name is usually the domain name where the resource record is found. Master file processing ends.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

System Action: The **named** server continues.

User or Operator Response: Correct the owner name for the directive in error.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9181I dns_master_load: *source_file*:*line_number*: **TTL** *ttl_value* **> MAXTTL, setting TTL to 0**

Explanation: The **named** server is processing a time-to-live (TTL) value and found the value to be greater than 2147483647 seconds. The **named** server will set this TTL value to zero and continues processing the master file.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

ttl_value value found for the TTL being processed.

System Action: The **named** server continues.

User or Operator Response: Correct the TTL value specified.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9182I dns_master_load: *source_file*:*line_number*: **no TTL specified**

Explanation: The **named** server is processing the master file class record time-to-live (TTL) and determined that the TTL referenced does not exist nor is there a default TTL established. Master file processing ends.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

System Action: The **named** server continues.

User or Operator Response: Specify a TTL value.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9183I dns_master_load: *source_file*:*line_number*: **using RFC 1035 TTL semantics**

Explanation: The default time-to-live (TTL) value was not specified by a $TTL directive. The name server will use the last explicitly stated value for the TTL. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9184I dns_master_load: *source_file*:*line_number*: **unknown RR type** *token*

Explanation: The **named** server found a master file option that is determined to be not implemented or is incorrect. This could be a typing error. Master file processing ends.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

token is token in the master file which is incorrect.

System Action: The **named** server continues.

User or Operator Response: Ensure that the option specified is correct.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9185I dns_master_load: *source_file*:*line_number*: **class** *class_specified* **!= zone class** *zone_class*

Explanation: The **named** server determined that the class specified in the master file record does not match the class of the zone.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

class_specified is the class specified.

zone_class is the zone class.

System Action: The **named** server continues.

User or Operator Response: Ensure that the class specified in the master file matches its zones class.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9186I dns_master_load: *source_file*:*line_number*: **TTL set to prior TTL (***ttl***)**

Explanation: The **named** server is setting the first time-to-live (TTL) resource data value to that which was specified in a prior TTL resource data value.

source_file is the name of the master file being processed.

line_number is the line number in master file.

ttl is the TTL whose value used.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9187I dns_master_load: *error_text*

Explanation: The **named** server failed to load the master file being processed because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the the TSO address space or region size and restart the **named** server.

Source Data Set: MASTER

Procedure Name: load

EZZ9188I dns_master_load: *source_file*:*line_number*: **ignoring out-of-zone data (***name***)**

Explanation: The **named** server found records that belong to another zone. This record is ignored and the master file processing continues.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

name is the DNS name of the resource record in error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the record in error is associated with the correct zone.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: generate

EZZ9189I *source_file*:*line_number*: **file does not end with newline**

Explanation: The **named** server is converting resource data from text into uncompressed wire form and found an end-of-file condition with the resource data being converted. The resource data might be incomplete.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

System Action: The **named** server continues.

User or Operator Response: Ensure that there are not end-of-file characters in the resource data specified.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_warneof

EZZ9190I dns_rdata_fromtext: *source_file*:*line_number*: **near eol:** *error_text*

Explanation: The **named** server is converting resource data from text into uncompressed wire form and found an end-of-line condition with the resource data being converted. The resource data might be incomplete.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that there are no end-of-line characters in the resource data specified.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_error

EZZ9191I **dns_rdata_fromtext:** *source_file*:*line_number*: **near eof:** *error_text*

Explanation: The **named** server is converting resource data from text into uncompressed wire form and found an end-of-file condition with the resource data being converted. The resource data might be incomplete.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that there are no end-of-file characters in the resource data specified.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_error

EZZ9192I **dns_rdata_fromtext:** *source_file*:*line_number*: **near** *number*: *error_text*

Explanation: The **named** server is converting resource data from text into uncompressed wire form and found an error condition where a number is incorrect.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

number is the token being processed that appears to be a number.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that there are no numeric characters in the resource data specified.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_error

EZZ9193I **dns_rdata_fromtext:** *source_file*:*line_number*: **near** *string*: *error_text*

Explanation: The **named** server is converting resource data from text into uncompressed wire form and found an error condition where a string or a quoted string is incorrect.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

string is the token being processed that appears to be a string or a quoted string.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that there are no character strings or quoted character strings in the resource data specified.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_error

EZZ9194I dns_rdata_fromtext: *source_file*:*line_number*: *error_text*

Explanation: The **named** server failed to convert resource data from text into uncompressed wire form.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Correct the specified resource data.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_error

EZZ9195I dns_rdata_fromtext: *source_file*:*line_number*: *error_text*

Explanation: The **named** server found an unexpected null token while converting resource data from text into uncompressed wire form.

source_file is the name of the master file containing the resource data being processed.

line_number is the line number in the master file.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Correct the specified resource data.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: fromtext_error

EZZ9196I options configuration failed: *error_text*

Explanation: The directory name specified by the directory option on the options statement in the **named** server configuration file is incorrect. The **named** server is parsing the **named** server configuration file and is unable to change directory to the directory specified in the directory option.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Ensure that the directory specified by the directory option is correct. See the *z/OS Communications Server: IP Configuration Reference* for information about the directory option.

System Programmer Response: None.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9197I Failed to create zone list

Explanation: The **named** server failed to create the zone list being parsed from the **named** server configuration file because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9198I error creating new zone

Explanation: The **named** server failed to create the new zone being parsed from the **named** server configuration file because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9199I error adding new zone to list

Explanation: The **named** server failed to add the newly created zone to the zone list because of insufficient memory while processing the zone element.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9200I zone configuration for *zone_name* **failed:** *error_text*

Explanation: The **named** server completed parsing the **named** server configuration file zone statements but detected an error condition while either configuring or reconfiguring a zone.

zone_name is the name of the zone being configured.

error_text describes the error.

System Action: If the *error_text* value is **already exists**, only the first occurence of *zone_name* will be loaded and the name server continues. For all other possible *error_text* values, the name server ends.

User or Operator Response: Correct the error described by the *error_text* and any prior error messages written to the SYSLOG DAEMON logging file.

System Programmer Response: None.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9201I Failed to create trusted key list

Explanation: The **named** server failed to process the trusted-keys variable in the **named** server configuration file because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9202I Failed to set trusted keys

Explanation: The **named** server failed to set the new trusted key in the trusted key list because it would duplicate an existing entry.

System Action: The **named** server ends.

User or Operator Response: Ensure that the trusted keys in the **named** server configuration file are unique.

System Programmer Response: None.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9203I No trusted key list defined!

Explanation: The **named** server could not find a trusted key in a trusted key list because the list did not exist in the current configuration or view.

System Action: The **named** server ends.

User or Operator Response: Ensure the trusted-keys statement in the **named** server configuration file is correct.

System Programmer Response: None.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9204I Failed to create trusted key

Explanation: The **named** server failed to create a new trusted key because of insufficient storage.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9205I Failed to append trusted key

Explanation: The **named** server failed to append a trusted key because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: yyparse

EZZ9206I dns_parse_namedconf: error creating mem context

Explanation: The **named** server failed to process the **named** server configuration file because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: dns_c_parse_namedconf

EZZ9208I dns_parse_namedconf: error creating config context

Explanation: The **named** server was not able to create the configuration context because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: dns_c_parse_namedconf

EZZ9209I dns_parse_namedconf: error creating lexer

Explanation: The **named** server was unable to create the lexer process to parse the **named** configuration file because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: CONFPARS

Procedure Name: dns_c_parse_namedconf

EZZ9210I *file_name*: **open:** *error_text*

Explanation: The **named** server failed to open the **named** server configuration file specified.

file_name is the name of **named** server configuration file.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Ensure that the **named** server file exists. Otherwise, contact system programmer to increase storage.

System Programmer Response: Increase the TSO address space or region size to resolve problems with insufficient memory.

Source Data Set: CONFPARS

Procedure Name: dns_c_parse_namedconf

EZZ9211I *file_name*:*line_number*: *message* **near** *token*

Explanation: The **named** server found semantic errors while parsing the **named** server configuration file.

file_name is the name of the **named** server configuration file.

line_number is the line number in the **named** server configuration file. It does not include comment lines.

message describes the error.

token is the token in error.

System Action: The **named** server might end depending on the severity of the error.

User or Operator Response: Correct the **named** configuration file statement in error. See the *z/OS Communications Server: IP Configuration Reference* for the proper syntax for the statements in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFPARS

Procedure Name: parser_complain

EZZ9212I *file_name*:*line_number*: *message*

Explanation: The **named** server found semantic errors while parsing the **named** server configuration file. This message will explain the failure found in the **named** server configuration file.

file_name is the name of the **named** server configuration file.

line_number is the line number in the **named** server configuration file. It does not include comment lines.

message describes the error.

System Action: The **named** server might end depending on the severity of the error.

User or Operator Response: Correct the **named** configuration file statement in error. See the *z/OS Communications Server: IP Configuration Reference* for the proper syntax for the statements in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFPARS

Procedure Name: parser_complain

EZZ9213I TCP/IP may not be active. named server, BIND v9 ending

Explanation: The **named** server cannot continue because TCP/IP Services is not active.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that TCP/IP Services is active and restart the **named** server.

Source Data Set: RESOLVER

Procedure Name: dns_resolver_create

EZZ9214I IBM message catalog problem. Using default messages

Explanation: The **named** server failed to retrieve a message from its message catalog. The message catalog, ns9.cat, might be corrupt. The file might have been transfered in ASCII but must be binary mode. The file is a binary and must be file transfered as BINARY.

System Action: The **named** server will use its default English messages.

User or Operator Response: Ensure that the message catalog, ns9.cat, has the correct file format and is the version distributed with BIND v9 DNS.

System Programmer Response: None.

Source Data Set: B9MSGCAT

Procedure Name: ibm_msgcat_init

EZZ9215I named, bind v9: Unable to run as non-swappable, errno=*description***, errno2=***errnojr*

Explanation: The **named** server attempted to run as non-swappable but was not allowed to by the S/390 RACF definitions. The **named** server will operate in a swappable environment.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: If you want the **named** server to run as non-swappable, then contact the systems programmer.

System Programmer Response: Contact the RACF Administrator to enable the **named** server to run non-swappable. See the *z/OS Communications Server: IP Configuration Reference* for the RACF requirements.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ9216I Running swappable

Explanation: The **named** server is running in a mode that allows it to be swapped out by the operating system.

System Action: The **named** server will operate in a swappable environment.

User or Operator Response: If you want the **named** server to run as non-swappable, then contact the systems programmer.

System Programmer Response: Contact the RACF Administrator to enable the **named** server to run non-swappable. See the *z/OS Communications Server: IP Configuration Reference* for the RACF requirements.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ9217I Running non-swappable

Explanation: The **named** server is running in a mode that prevents it from being swapped out by the operating system.

System Action: The **named** server will operate in a non-swappable environment.

User or Operator Response: If you want the **named** server to run as swappable, then contact the systems programmer.

System Programmer Response: Contact the RACF Administrator to enable the **named** server to run swappable. See the *z/OS Communications Server: IP Configuration Reference* for the RACF requirements.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ9218I No valid *cpu_configuration* data returned from CSRSI

Explanation: The CPU hardware is not able to return the number of logical CPUs.

cpu_configuration is the mode of operation of the z/OS system, which can be **VM**, **LPAR**, or **MACHINE**.

System Action: The **named** server will operate using a default of two CPUs.

User or Operator Response: If the number of logical CPUs is known for this CPU configuration, then stop and restart the name server with the **-n** option and specify the number of CPUs manually.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9219I Unknown variety returned from CSRSI

Explanation: The CPU hardware is not able to return the number of logical CPUs.

System Action: The **named** server will operate using a default of two CPUs.

User or Operator Response: If the number of logical CPUs is known for this CPU configuration, then stop and restart the name server with the **-n** option and specify the number of CPUs manually.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9220I Processor model does not support CSRSI

Explanation: The CPU hardware is not able to return the number of logical CPUs.

System Action: The **named** server will operate using a default of two CPUs.

User or Operator Response: If the number of logical CPUs is known for this CPU configuration, then stop and restart the name server with the **-n** option and specify the number of CPUs manually.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: set_cpu_number

EZZ9221I Unable to write to file, logging halted

Explanation: The **named** server failed to write log messages to a configured log because of insufficient disk space.

System Action: The **named** server continues.

User or Operator Response: Ensure that there is enough space for the logging files defined in the **named** configuration file. Either alter the space definition in the **named** configuration file or contact the system programmer for more space. After the space problem is corrected, stop and re-start the **named** server or reload the **named** server to resume logging.

System Programmer Response: Ensure that there is no problem with either the DASD or the HFS the log files reside on. Otherwise, allocate more disk space for the **named** server log files.

Source Data Set: ISCLOG

Procedure Name: isc_log_doit

EZZ9222I forward zones do not have a notify-source-v6 field

Explanation: The option notify-source-v6, specified in the **named** configuration file, is incorrect in a forward zone. The option notify-source-v6 is only allowed in slave zones.

System Action: The **named** server ends.

User or Operator Response: Remove the notify-source-v6 option from any forward zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setnotifysourcev6

EZZ9223I stub zones do not have a notify-source-v6 field

Explanation: The option notify-source-v6, specified in the **named** configuration file, is incorrect in a stub zone. The option notify-source-v6 is only allowed in slave zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the notify-source-v6 option from any stub zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotifysourcev6

EZZ9224I hint zones do not have a notify-source-v6 field

Explanation: The option notify-source-v6, specified in the **named** configuration file, is incorrect in a hint zone. The option notify-source-v6 is only allowed in slave zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the notify-source-v6 option from any hint zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotifysourcev6

EZZ9225I forward zones do not have a notify-source-v6 field

Explanation: The option notify-source-v6, specified in the **named** configuration file, is incorrect in a forward zone. The option notify-source-v6 is only allowed in slave zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the notify-source-v6 option from any forward zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getnotifysourcev6

EZZ9226I hint zones do not have a min-retry-time field

Explanation: The option min-retry-time, specified in the **named** configuration file, is incorrect in a hint zone. The option min-retry-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the min-retry-time option from any hint zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setminretrytime

EZZ9227I forward zones do not have a min-retry-time field

Explanation: The option min-retry-time, specified in the **named** configuration file, is incorrect in a forward zone. The option min-retry-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the min-retry-time option from any forward zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setminretrytime

EZZ9228I hint zones do not have a min-retry-time field

Explanation: The option min-retry-time, specified in the **named** configuration file, is incorrect in a hint zone. The option min-retry-time is only allowed in master, slave, or stub zones.

System Action: If the **named** is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the min-retry-time option from any hint zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getminretrytime

EZZ9229I forward zones do not have a min-retry-time field

Explanation: The option min-retry-time, specified in the **named** configuration file, is incorrect in a forward zone. The option min-retry-time is only allowed in master, slave, or stub zones.

System Action: If the **named** is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the min-retry-time option from any forward zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getminretrytime

EZZ9230I hint zones do not have a max-retry-time field

Explanation: The option max-retry-time, specified in the **named** configuration file, is incorrect in a hint zone. The option max-retry-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-retry-time option from any hint zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxretrytime

EZZ9231I forward zones do not have a max-retry-time field

Explanation: The option max-retry-time, specified in the **named** configuration file, is incorrect in a forward zone. The option max-retry-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-retry-time option from any forward zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxretrytime

EZZ9232I hint zones do not have a max-retry-time field

Explanation: The option max-retry-time, specified in the **named** configuration file, is incorrect in a hint zone. The option max-retry-time is only allowed in master, slave, or stub zones. This option is ignored.

System Action: If the **named** is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the max-retry-time option from any hint zones in the **named** server configuration file.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxretrytime

EZZ9233I error message would overflow

Explanation: The **named** server failed to generate an error message during the **named** server configuration file processing because the message is too big.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The LE dump produced when re-creating this error.

Source Data Set: CONFPARS

Procedure Name: parser_complain

EZZ9235I isc_mutex_init failed in new_adbfind()

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ADB

Procedure Name: new_adbfind

EZZ9236I handle_signal() *signal* **setup:** *description*

Explanation: The **named** server received a signal that it could not handle.

signal is a z/OS UNIX System Services signal. The valid signals handled by the **named** server are as follows:

- SIGHUP
- SIGINT
- SIGPIPE
- SIGTERM

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the correct signal was passed issued to the **named** server.

System Programmer Response: None.

Source Data Set: APP

Procedure Name: handle_signal

EZZ9238I isc_app_start() sigsetops: *description*

Explanation: The **named** server failed to set up a signal set and mask because an incorrect was specified. The **named** server handles the following signals:

- SIGHUP
- SIGINT
- SIGTERM

description describes the error.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The LE dump produced when re-creating this error.

Source Data Set: APP

Procedure Name: isc_app_start

EZZ9239I isc_app_start() pthread_sigmask: *description*

Explanation: The **named** server failed to set the blocked signals for a thread because the signals in the mask are incorrect.

description describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.

Source Data Set: APP

Procedure Name: isc_app_start

EZZ9240I isc_app_run() sigsetops: *description*

Explanation: The **named** server failed to set up the signal set and masks because of an incorrect signal to block was specified. The **named** server blocks the following signals:

- SIGHUP
- SIGINT
- SIGIOERR
- SIGTERM

description describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.

Source Data Set: APP

Procedure Name: isc_app_run

EZZ9243I isc_app_shutdown() kill: *description*

Explanation: The **named** server failed to shutdown because a thread could not be ended.

description describes the error.

System Action: The **named** server does not end and is unuseable.

User or Operator Response: Contact the system programmer.

System Programmer Response: Issue the z/OS UNIX kill -9 command to end the **named** server. Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The dump of the **named** address space.

Source Data Set: APP

Procedure Name: isc_app_shutdown

EZZ9245I isc_app_reload() kill: *description*

Explanation: The **named** server failed to reload because a thread could not be ended.

description describes the error.

System Action:

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The dump of the **named** address space taken when the problem was re-created.

Source Data Set: APP

Procedure Name: isc_app_reload

EZZ9246I unknown auth algorithm *algorithm_returned*

Explanation: The **named** object manager returned an unknown algorithm while attempting to make a key. The only algorithm allowed by the object manager is hmac-md5. The object manager will not be able to register this key.

algorithm_returned is the algorithm name returned.

System Action: The **named** server continues.

User or Operator Response: Ensure that the algorithm associated with the security key is specified as **hmac-md5**. See the *z/OS Communications Server: IP Configuration Reference* for more information about specifying algorithms for keys.

System Programmer Response: None.

Source Data Set: AUTH

Procedure Name: auth_makekey

EZZ9247I isc_mutex_init failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DISPATCH

Procedure Name: qid_allocate

EZZ9248I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DNSCACHE

Procedure Name: dns_cache_create

EZZ9249I isc_task_create() failed: *error_text*

Explanation: The **named** server failed to create the cache cleaner task because of insufficient memory or the **named** server is shutting down.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: If the error is insufficient memory, then contact the system programmer to increase memory.

System Programmer Response: Increase the the TSO address space or region size and restart the **named** server.

Source Data Set: DNSCACHE

Procedure Name: cache_cleaner_init

EZZ9250I cache cleaner: isc_task_onshutdown() failed: *error_text*

Explanation: The **named** server failed because of insufficient memory.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: If the error is insufficient storage, then contact the system programmer to increase memory.

System Programmer Response: Increase the the TSO address space or region size and restart the **named** server.

Source Data Set: DNSCACHE

Procedure Name: cache_cleaner_init

EZZ9251I isc_timer_create() failed: *error_text*

Explanation: The **named** server failed to create a cache cleaner task timer because of insufficient memory or the **named** server is not able to obtain the time of day.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the z/Series host is able to return the the correct time. Use the z/OS UNIX **date** command to ensure that the CPUs date is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command. If the error is insufficient memory, then contact the system programmer.

System Programmer Response: Increase the the TSO address space or region size and restart the **named** server.

Source Data Set: DNSCACHE

Procedure Name: cache_cleaner_init

EZZ9252I cache cleaner: dns_dbiterator_first() failed: *error_text*

Explanation: The **named** server cache cleaner failed to clean the cache. The information reflected by the cache will become stale and you will experience erroneous results.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Shut down and restart the **named** server to rebuild the cache.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: begin_cleaning

EZZ9253I cache cleaner: dns_dbiterator_current() failed: *error_text*

Explanation: The **named** server failed to perform incremental cache cleaning. The **named** server will stop cleaning the cache and set the cache cleaner to an idle state. The information reflected by the cache will become stale and you will experience erroneous results.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Shut down and restart the **named** server to rebuild the cache.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: incremental_cleaning_action

EZZ9254I cache cleaner: dns_db_expirenode() failed: *error_text*

Explanation: The **named** server cache cleaner failed to determine which nodes in the cache are expired while performing intermediate cache cleaning. The information reflected by the cache will become stale and you will experience erroneous results. The **named** server cache cleaner continues.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Shut down and restart the **named** server to rebuild the cache.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: incremental_cleaning_action

EZZ9255I cache cleaner: dns_dbiterator_next() failed: *error_text*

Explanation: The **named** server cache cleaner failed to determine the next database node to clean while performing intermediate cache cleaning. The information reflected by the cache will become stale and you will experience erroneous results.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Shut down and restart the **named** server to rebuild the cache.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: incremental_cleaning_action

EZZ9256I cache cleaner: dns_dbiterator_pause() failed: *error_text*

Explanation: The **named** server failed to perform intermediate cache cleaning. The information reflected by the cache will become stale and you will experience erroneous results.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Shut down and restart the **named** server to rebuild the cache.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: incremental_cleaning_action

EZZ9257I cache cleaner: dns_db_expirenode() failed: *error_text*

Explanation: The **named** server failed to perform intermediate cache cleaning. The information reflected by the cache will become stale and you will experience erroneous results. The **named** server cache cleaner continues.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Shut down and restart the **named** server to rebuild the cache.

System Programmer Response: None.

Source Data Set: DNSCACHE

Procedure Name: dns_cache_clean

EZZ9258I isc_socket_accept() failed: *error_text*

Explanation: The **named** server failed to accept a socket from the client listener because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the the TSO address space or region size and restart the **named** server.

Source Data Set: CLIENT

Procedure Name: client_accept

EZZ9259I pthread_cond_timedwait() returned *error_text*

Explanation: A **named** server thread failed a time out.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The dump of the **named** address space.

Source Data Set: CONDITIO

Procedure Name: isc_condition_waituntil

EZZ9260I unknown type in omapi_connection_putdata: *data_type*

Explanation: The **named** object manager detected an internal error.

data_type is the data type.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The dump of the **named** address space.

Source Data Set: CONNECTI

Procedure Name: omapi_connection_putdata

EZZ9261I Unexpected path to connection_destroy - the connection object was dereferenced without a previous disconnect

Explanation: The **named** object manager connection being destroyed was not previously disconnected. The **named** object manager will disconnect the connection being destroyed.

System Action: The **named** server will continue.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CONNECTI

Procedure Name: connection_destroy

EZZ9262I unknown type in omapi_data_create: *data_type*

Explanation: The **named** object manager detected an internl error.

data_type is the data type

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Dump the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The dump of the **named** address space.

Source Data Set: OMAPDATA

Procedure Name: omapi_data_create

EZZ9263I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.`

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: MASTER

Procedure Name: loadctx_create

EZZ9264I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TSIG

Procedure Name: dns_tsigkey_createfromkey

EZZ9265I isc_mutex_init failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DISPATCH

Procedure Name: dispatch_allocate

EZZ9266I isc_result_register() failed: *error_code*

Explanation: The **named** server failed to build the result table because of insufficient memory.

error_code describes the error and will be a 1 for insufficient memory.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: DST_RESU

Procedure Name: initialize_action

EZZ9267I **fcntl(***file_descriptor***, F_SETFL,** *flags***):** *description*

Explanation: The **named** server detected an internal error.

file_descriptor is the file descriptor for the /dev/random device.

flags is the O_NONBLOCK flag being set for the file descriptor.

description describes the error.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends when the **named** server fails to create a socket manager. Otherwise, the **named** server continues.

User or Operator Response: If the **named** server ends, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** server configuration file to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The component trace requested. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.
- The LE dump produced by this error.

Source Data Set: ENTROPY

Procedure Name: make_nonblock

EZZ9268I **making interface scan socket:** *description***, errno2=***errnojr*

Explanation: The **named** server detected an internal error.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **IOCTL** option and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Dump the **named** address space if the **named** server did not end. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.

Source Data Set: INTRFCEI

Procedure Name: isc_interfaceiter_create

EZZ9269I *file***:***line***: fatal error:**

Explanation: The **named** server uses this message to report its fatal errors.

file is the file name.

line is the line number in the file.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Review the previous error messages written to the SYSLOG DAEMON logging file or the **named** logging files and fix them. If you are not able to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.

Source Data Set: ISCERROR

Procedure Name: default_fatal_callback

EZZ9270I Incorrect DNS_VERSION environment variable value

Explanation: The DNS-VERSION environment variable was set to an incorrect value. The supported values are as follows:

- **v9** for BIND v9 DNS
- **v4** for BIND v4 DNS

System Action: The **named** server ends.

User or Operator Response: Ensure that the DNS_VERSION environment variable being used is correct.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ9271I -V option value, *version_command_line_argument* **, is not valid**

Explanation: The **named** server cannot use the value specified by the **-V** command line argument. The **named** server will generate a command line usage diagram in response to the incorrect **-V** option value. The correct values are:

- **v9** for BIND v9 DNS
- **v4** for BIND v4 DNS

version_command_line_argument is the value specified by the **-V** command line variable.

System Action: The **named** server ends.

User or Operator Response: Specify a valid **-V** command line variable.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ9272I named, BIND v4 options specified for named, BIND v9

Explanation: BIND v4 command line arguments were specified while starting the BIND v9 **named** server.

System Action: The **named** server ends.

User or Operator Response: Reveiw command line arguments specified to determine which are associated with BIND v4 and remove them.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: version_check

EZZ9273I named, BIND v9 options specified for named, BIND v4

Explanation: BIND v9 command line arguments were specified while starting the BIND v4 **named** server.

System Action: The **named** server ends.

User or Operator Response: Reveiw command line arguments specified to determine which are associated with BIND v9 and remove them.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: version_check

EZZ9274I illegal mixture of named, BIND v4 and v9 options

Explanation: A mixture of BIND v4 and BIND v9 command line variables were specified when starting the **named** server.

System Action: The **named** server ends.

User or Operator Response: Specify the correct command line variables for the appropriate BIND DNS version.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: version_check

EZZ9276I named, BIND v9 ending

Explanation: BIND v9 does not have superuser authority.

System Action: The **named** server ends.

User or Operator Response: Ensure that the **named** server user ID has superuser capabilities.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: check_user_authorization

EZZ9277I unknown datatype in omapi_data_dereference: *type*

Explanation: The **named** server detected an internal error.

type is the incorrect data.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Inform the user of the **rndc** client that the **rndc** client being used might not be downward compatible with the **named** server software. Temporarily disable this client through the **controls** address match list in the **named** server configuration file, or completely disable **rndc** by removing the **controls** statement from the **named** server configuration file. Any changes to the **named** server configuration file must be followed by an **rndc reload** or SIGHUP signal to re-read the **named** server configuration file.

System Programmer Response: None.

Source Data Set: OMAPDATA

Procedure Name: omapi_data_dereference

EZZ9278I Unexpected state *state*

Explanation: The **named** server could not determine the state of the token being processed by lex. This error can occur while parsing the following files:

- the **named** server configuration file
- the **rndc** configuration file

- a master file
- resource data

The token being processed is not one of the following:

- carriage return / line feed
- string
- number
- comment
- end-of-line
- end-of-file
- double quote

state is the state of the lexer.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: LEX

Procedure Name: isc_lex_gettoken

EZZ9279I Unexpected state *state*

Explanation: The **named** server failed to convert a DNS name from text format into uncompressed wire format. The DNS name being converted is incorrect.

state is the state of the data being converted.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Ensure all DNS names are valid names and are properly typed in the **named** server configuration file. If you are not able to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: DNSNAME

Procedure Name: dns_name_fromtext

EZZ9280I Unexpected label type *label_type*

Explanation: The **named** server failed to convert a DNS name from uncompressed wire format to text format. The DNS name in uncompressed wire format has an incorrect label.

label_type is the master file internal record format.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: DNSNAME

Procedure Name: dns_name_totext

EZZ9281I Unexpected label type *label_type*

Explanation: The **named** server failed to convert a DNS name to lower case characters.

label_type is the master file internal record format.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: DNSNAME

Procedure Name: dns_name_downcase

EZZ9282I Unknown state *state*

Explanation: The **named** server failed to convert a DNS name from wire format to text.

state is the state of the data in error.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Ensure all DNS names are valid names and are properly keyed entered in the **named** server configuration file. If you are not able to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.

- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: DNSNAME

Procedure Name: dns_name_fromwire

EZZ9284I unknown address family: *family*

Explanation: The **named** server failed to process a socket address because of an unknown protocol family.

family is the protocol family in error.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Ensure all socket addresses defined are of the Internet protocol family. If you are not able to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: SOCKADDR

Procedure Name: isc_sockaddr_pf

EZZ9285I unknown address family: *family*

Explanation: The **named** server failed to set the port for a socket address because of an unknown socket address protocol family.

family is the protocol family in error.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Ensure all socket addresses defined are of the Internet protocol family. If you are not able to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: SOCKADDR

Procedure Name: isc_sockaddr_setport

EZZ9286I unknown address family: *family*

Explanation: The **named** server failed to get a port for a socket address because of an unknown socket address protocol family.

family is the protocol family in error.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Ensure all socket addresses defined are of the Internet protocol family. If you are not able to resolve this problem, then contact the system programmer.

System Programmer Response: Obtain the following documentation and contact the IBM Software Support Center:

The **named** server configuration file.

The SYSLOG DAEMON logging file.

All configured **named** logs.

Use **rndc dumpdb** to dump the **named** servers cache to a file

Setup a component trace for the SYSTCPIP component. See the *z/OS Communications Server: IP Diagnosis* for instructions on setting and producing a component trace.

The LE dump produced when re-creating this error.

Source Data Set: SOCKADDR

Procedure Name: isc_sockaddr_getport

EZZ9287I write() failed during watcher poke: *description*, **errno2=***errnojr*

Explanation: The **named** server failed to process a poke because of a problem writing to a file descriptor.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: ISCSOCKT

Procedure Name: select_poke

EZZ9288I read() failed during watcher poke: *description*, **errno2=***errnojr*

Explanation: The **named** server failed to read an internal file descriptor.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: ISCSOCKT

Procedure Name: select_readmsg

EZZ9289I select() failed: *description*, **errno2=***errnojr*

Explanation: The **named** server watcher failed. The socket manager failed to use a file descriptor.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: ISCSOCKT

Procedure Name: watcher

EZZ9290I RUNTIME_CHECK(*expression***) failed**

Explanation: The **named** server failed because of a runtime check condition.

expression is the reason **named** server received a runtime check.

System Action: A Language Environment (LE) CEE dump was taken and the **named** server ends.

User or Operator Response: Review the failure returned in this message and any prior messages to resolve the problem. If you are not able to resolve the problem then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Use **rndc dumpdb** to dump the **named** servers cache to a file.
- The LE dump produced when re-creating this error.

Source Data Set: ISCERROR

Procedure Name: isc_error_runtimecheck

EZZ9291I RESOLVER INITIALIZATION COMPLETE

Explanation: The Resolver address space initialized and is ready to accept MODIFY and STOP commands, and Resolver services are available to applications.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBREINI

Procedure Name: EZBREINI

EZZ9292I RESOLVER ENDING

Explanation: A STOP command was entered. The Resolver address space is ending normally.

System Action: Resolver services are no longer available to applications.

User or Operator Response: The Resolver address space can be restarted to make Resolver services available to applications.

System Programmer Response: None.

Source Data Set: EZBREINI

Procedure Name: EZBREINI

EZZ9293I cmdname COMMAND PROCESSED

Explanation: The Resolver address space accepted and processed the MODIFY RESOLVER,cmdname command.

cmdname will be one of the following:

DISPLAY
Display the current settings for the GLOBALTCPIPDATA file and the DEFAULTTCPIPDATA file.

REFRESH
Notify the Resolver that changes to the Resolver setup file, to the GLOBALTCPIPDATA file, or to the DEFAULTTCPIPDATA have been made. Subsequent requests for Resolver services from applications will be processed using those changes.

System Action: The Resolver address space waits for another MODIFY or STOP command.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBREINI, EZBRECFG

Procedure Name: EZBREINI, EZBRECFG

EZZ9294I INCORRECT *type* SYNTAX

Explanation: The MODIFY RESOLVER command entered did not have correct syntax.

type will be one of the following:

COMMAND
The syntax of the command is not correct.

FILENAME
The syntax for the filename specified as the setup file is not correct.

System Action: The MODIFY RESOLVER command is ignored.

User or Operator Response: Correct the MODIFY RESOLVER command and reenter it. See the *z/OS Communications Server: IP System Administrator's Commands* for more information about Resolver commands.

System Programmer Response: None.

Source Data Set: EZBREINI, EZBRECFG

Procedure Name: EZBREINI, EZBRECFG

EZZ9295I INCORRECT STATEMENT SYNTAX ON LINE *lineno*

Explanation: Line *lineno* in the Resolver setup file did not have the correct syntax.

System Action: The incorrect setup file statement is rejected. The Resolver address space ends.

User or Operator Response: Contact your system programmer.

System Programmer Response: Correct the setup file statement and restart the Resolver address space. See the *z/OS Communications Server: IP Configuration Reference* for more information about Resolver setup file statements.

Source Data Set: EZBRECFG

Procedure Name: EZBRECFG

EZZ9296I UNABLE TO ACCESS FILE *filename* **- RC** *rc*

Explanation: The Resolver was unable to access the MVS data set or the HFS file *filename* while processing a MODIFY RESOLVER,REFRESH command.

rc is the return code and will be in one of the following:

Return code (hexadecimal)	**Meaning**	**Corrective action for //DD:SETUP**	**Corrective action for** *filename*
00080004	DDNAME not allocated.	Not applicable.	Filename is in the format "//DD:ddname". Ensure that the specified DDNAME exists.
00080008	Open failed.	Ensure that the specified setup file exists and is accessible to the Resolver.	Ensure that the specified MVS data set or HFS file exists and is accessible to the application.
0008000C	Record format of MVS data set is not supported.	Ensure that the specified MVS data set has record format F or FB.	Ensure that the specified MVS data set has record format F or FB.
00080010	Data set in use.	Not applicable.	Ensure that the specified MVS data set is not allocated by another job.
00080024	Insufficient storage to allocate and open file.	Increase region size for Resolver.	Increase region size for application.
0010**xxxx**	Dynamic allocation failed.	Not applicable.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Guide* and perform the corrective actions listed in the DYNALLOC error reason code table for SVC99 error code xxxx, and reenter the MODIFY RESOLVER,REFRESH command.

System Action: The MODIFY RESOLVER,REFRESH command is ignored.

EZZ9297E

User or Operator Response: Contact your system programmer. For example, if an HFS file is specified and a RC=00080008, the JESMSGLG may have the message OMVS SEGMENT NOT DEFINED. In which case, the Resolver address space will not have HFS file access until an OMVS segment is defined for it by your system programmer.

System Programmer Response: See Corrective Actions listed above.

Source Data Set: EZBRECFG

Procedure Name: EZBRECFG

EZZ9297E UNABLE TO ACCESS FILE *filename* **- RC** *rc*

Explanation: If *filename* is //DD:SETUP, the Resolver was unable to access the MVS data set or the HFS file defined on the //SETUP DD statement in the Resolver JCL procedure. If *filename* is an MVS data set name or an HFS file name, the Resolver is unable to access the MVS data set or the HFS file while processing Resolver setup file statements.

rc is the return code and will be in one of the following:

Return code (hexadecimal)	**Meaning**	**Corrective action for //DD:SETUP**	**Corrective action for** *filename*
00080004	DDNAME SETUP was not allocated.	Add //SETUP DD to Resolver JCL procedure.	Not applicable
00080008	Open failed.	Ensure that the specified setup file exists and is accessible to the Resolver.	Ensure that the specified MVS data set or HFS file exists and is accessible to the Resolver.
00080010	Data set in use.	Not applicable.	Ensure that the specified MVS data set is not allocated by another job.
0008000C	Record format of MVS data set is not supported.	Ensure that the specified MVS data set has record format F or FB.	Ensure that the specified MVS data set has record format F or FB.
00080024	Insufficient storage to allocate and open file.	Increase region size for Resolver.	Increase region size for Resolver.
0010**xxxx**	Dynamic allocation failed.	Not applicable.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Guide* and perform the corrective actions listed in the DYNALLOC error reason code table for SVC99 error code **xxxx**, and restart the Resolver.

System Action: The Resolver address space ends.

User or Operator Response: None.

System Programmer Response: See Corrective Actions listed above.

Source Data Set: EZBRECFG

Procedure Name: EZBRECFG

EZZ9298I *tcpipdatafiletype/ipnodesfiletype* **-** *filename*

Explanation: This message is issued in response to a MODIFY RESOLVER,DISPLAY command or a MODIFY RESOLVER,REFRESH command.

If *tcpipdatafiletype* is DEFAULTTCPIPDATA, then *filename* is the MVS data set or the HFS file that is currently in use as the default TCPIP.DATA file. If *tcpipdatafiletype* is GLOBALTCPIPDATA, then *filename* is the MVS data set or the HFS file that is currently in use as the global TCPIP.DATA file.

If *ipnodesfiletype* is GLOBALIPNODES, then *filename* is the MVS data set or the HFS file that is currently in use as the global IPNODES file. If *ipnodesfiletype* is DEFAULTIPNODES, then *filename* is the MVS data set or the HFS file that is currently in use as the default IPNODES file.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBREINI, EZBRECFG

Procedure Name: EZBREINI, EZBRECFG

EZZ9299E RESOLVER INITIALIZATION FAILED - RC *rc* **RSN** *rsn*

Explanation: The Resolver address space was unable to initialize.

rc is the return code and *rsn* is the reason code and they will be in one of the following combinations. When *rc* is 16, the **xx** in the reason code is the Authorized Assembler Services return code. To look up this return code, go to the location shown in the Corrective Action below.

Return code (decimal)	Reason code (hexadecimal)	Explanation	Corrective action
8	000D0000	Resolver setup file error.	Perform corrective actions for Resolver messages regarding setup file erors, and restart the Resolver.
12	00000000	MVS PPT entry for EZBREINI is missing or incorrect.	Ensure that IEFSDPPT was correctly installed, and that there have been no SET SCH=xx commands entered that would override it, and restart the Resolver.

EZZ9299E

Return code (decimal)	Reason code (hexadecimal)	Explanation	Corrective action
16	000200**xx**	Unable to establish ESTAE recovery routine.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG* and perform the corrective actions listed in the return and reason codes table for ESTAEX, and restart the Resolver.
	000300**xx**	Unable to locate EZBRESRV service routine load module.	Refer to the *z/OS DFSMS Macro Instructions for Data Sets* and perform the corrective actions listed in the completion codes table for BLDL, and restart the Resolver.
	000400**xx**	Unable to obtain CSA storage for the main Resolver control block, or unable to obtain CSA storage for the service routine load module.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO* and perform the corrective actions listed in the return and reason codes table for STORAGE OBTAIN, and restart the Resolver.
	000500**xx**	Unable to add service routine entry points to dynamic LPA.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* and perform the corrective actions listed in the return and reason codes table for CSVDYLPA ADD, and restart the Resolver.
	000600**xx**	Unable to load EZBRESRV service routine load module into CSA.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU* and perform the corrective actions listed in the return and reason codes table for LOAD, and restart the Resolver.
	000700**xx**	Unable to set system AX.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* and perform the corrective actions listed in the return codes table for AXSET, and restart the Resolver.
	000800**xx**	Unable to create service routine PC entry table.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG* and perform the corrective actions listed in the return codes table for ETCRE, and restart the Resolver.
	000900**xx**	Unable to obtain system LX.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU* and perform the corrective actions listed in the return codes table for LXRES, and restart the Resolver.
	000A00**xx**	Unable to connect service routine PC entry table to system entry tables.	Refer to the *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG* and perform the corrective actions listed in the return codes table for ETCON, and restart the Resolver.

System Action: The Resolver address space ends.

User or Operator Response: Contact your system programmer.

System Programmer Response: See Corrective Actions listed above.

Source Data Set: EZBREINI

Procedure Name: EZBREINI

EZZ9301E RESOLVER ENDED ABNORMALLY

Explanation: An unrecoverable error occurred in the Resolver address space.

System Action: An SVC dump of the Resolver address space is created and the Resolver address space ends.

User or Operator Response: Restart the Resolver address space.

System Programmer Response: Save the SVC dump and contact the IBM Software Support Center.

Source Data Set: EZBREINI

Procedure Name: EZBREINI

EZZ9302I UNABLE TO ACCESS TRACE FILE *filename* **RC** *rc*

Explanation: The resolver trace output file could not be opened for writing.

filename is specified by way of the SYSTCPT DD card or the RESOLVER_TRACE environment variable.

rc is the return code and will be one of the following:

Return code (hexadecimal)	**Meaning**	**Corrective action for** *filename*
00080008	OPEN failed	Ensure that the specified MVS data set or HFS file exists and is accessible to the user or job.
0008000C	Record format of MVS data set is not supported	Ensure that the specified MVS data set has record format F or FB and LRECL no larger than 80.
00080010	data set currently in use by another user or job	Either specify a different filename or stop the user or job currently using the MVS data set or HFS file.
00080024	Insufficient storage to allocate and open file	Increase region size for user or job.
0010**xxxx**	Dynamic allocation failed	Refer to the *z/OS MVS Programming: Authorized Assembler Services Guide* and perform the corrective actions listed in the DYNALLOC error reason code table for SVC99 error code **xxxx**.

System Action: The user or job continues. No resolver traces are written.

User or Operator Response: See corrective actions listed above.

System Programmer Response: None

Source Data Set: EZBRERIN

Procedure Name: EZBRERIN

EZZ9303I RESOLVER INITIALIZATION FAILED - RESOLVER ALREADY ACTIVE

Explanation: The Resolver was already started. Only one resolver address space may be active.

System Action: This RESOLVER fails to initialize. The currently active resolver continues to operate.

User or Operator Response: Either stop the currently active resolver or issue the MODIFY RESOLVER,REFRESH command to specify and activate a different setup file. See the *z/OS Communications Server: IP System Administrator's Commands* for more information about resolver commands.

System Programmer Response: None

Source Data Set: EZBREINI

Procedure Name: EZBREINI

EZZ9304I *resolversetupstatement*

Explanation: This message is issued in response to a MODIFY RESOLVER,DISPLAY command or a MODIFY RESOLVER,REFRESH command.

resolversetupstatement is one of the following:

COMMONSEARCH
Indicates that the search order for the local host file will be the same for either an IPv6 or an IPv4 name query.

NOCOMMONSEARCH
Indicates that the search order for the local host file will be different for an IPv6 or an IPv4 name query.

See the *z/OS Communications Server: IP Configuration Guide* for more information about resolver search order.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBREINI, EZBRECFG

Procedure Name: EZBREINI, EZBRECFG

EZZ9314E TCP/IP WAITING FOR OMVS TO INITIALIZE

Explanation: OMVS is not initialized. TCP/IP cannot complete its initialization until OMVS initialization is complete.

System Action: TCP/IP is waiting until OMVS completes initialization.

User or Operator Response: If OMVS has been SHUTDOWN (using the MODIFY OMVS command), then restart OMVS. If OMVS initialization (during IPL) is not complete, TCP/IP will resume initialization when the OMVS initialization is complete.

System Programmer Response: None.

Source Data Set: EZBTIINI

Procedure Name: EZBTIINI

EZZ9315E TCP/IP WAITING FOR RESOLVER TO INITIALIZE

Explanation: The Resolver is not initialized. TCP/IP cannot complete its initialization until Resolver initialization is complete.

System Action: TCP/IP is waiting until Resolver initialization completes.

User or Operator Response: Start the Resolver to allow TCP/IP initialization to continue, or issue the FORCE ARM command to end TCP/IP.

System Programmer Response: None.

Source Data Set: EZBTIINI

Procedure Name: EZBTIINI

EZZ9316I TRMD TCP statistics:*timestamp*, **port=***port*, **host=***host*, **lhost=***lhost*, **peak=***peak*, **hostpeak=***hostpeak*, **requests=***requests*, **terminations=***terminations*, **current=***current*, **suggested_limit=***suggested_limit*, **suggested_percentage=***suggested_percentage*, **duration=***duration*, **warnings=***warnings*, **qos_exceptions=***qos_exceptions*, **action=***action*

Explanation: Traffic Regulation statistics have been gathered.

timestamp is the time the statistics were recorded.

port is the port from which these statistics were collected.

host is the IP address that requested a connection and held the maximum number of concurrent connections for a single host during the collection interval.

lhost is the IP address of the local host that held the maximum number of concurrent connections for a single host during the collection interval.

peak is the maximum number of concurrent connections to this port held during the collection interval.

hostpeak is the largest number of connections held by host at any time during the collection interval.

requests is the number of connections requested during the collection interval.

terminations is the number of connections that were disconnected during the collection interval.

current is the number of connections currently active on this port.

suggested_limit is a suggested value for TotalConnections based on the traffic during this collection interval. If TypeActions Limit is specified, the value will be 0.

suggested_ percentage is a suggested value for Percentage based on the traffic during this collection interval. If TypeActions Limit is specified, the value will be 0.

duration is the amount of time spent in a constrained state. A constrained state occurs when the number of available connections dropped to 10% of TotalConnections value. A constrained state ends when the number of available connections increases to 12% of TotalConnections.

warnings is the number of connections that would have been refused if TypeActions Limit had been specified.

qos_exceptions is the number of connections that would have been refused by Intrusion Detection Services (IDS) policy, but allowed by QoS policy if TypeActions Limit had been specified.

action is the policy TypeAction indicator. LIMIT indicates TypeActions LIMIT was specified in the policy; NOLIMIT indicates TypeActions LIMIT was specified in the policy.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteStatEntries

EZZ9317I TRMD TCP QOS exception made:*timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **host_current=***host_current*, **qos_limit=***qos_limit*

Explanation: The Intrusion Detection Services (IDS) policy for TCP specified TypeActions Limit. A connection that did not meet current IDS policy specification was allowed by QoS policy.

timestamp is the date and time at which the QoS exception was made.

lhost is the IP address of the local host.

port is the port on which the QoS exception occurred.

host is the host for which the QoS exception was made.

available is the number of connections out of the total pool that were still available.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

host_current is the number of connections that the host held at the time of the QoS exception.

qos_limit is the number of connections allowed by the excepting QoS policy.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9318I **TRMD TCP QOS exception logged:***timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **host_current=***host_current*, **qos_limit=***qos_limit*

Explanation: The Intrusion Detection Services (IDS) policy for TCP did not specify TypeActions Limit. A connection that would have been outside current IDS policy specification but was allowed by QoS policy was established.

timestamp is the date and time at which the QoS exception was logged.

lhost is the IP address of the local host.

port is the port on which the QoS would have exception occurred.

host is the host for which the QoS exception would have been made.

available is the number of connections available out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

host_current is the number of connections that the host held at the time of the QoS exception.

qos_limit is the number of connections allowed by the excepting QoS policy.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9319I **TRMD TCP connection would have been refused:***timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **host_current=***host_current*

Explanation: The Intrusion Detection Services (IDS) policy for TCP did not specify TypeActions Limit. A connection was established that would have been outside current IDS policy specificaton.

timestamp is the date and time at which the connection would have been refused.

lhost is the IP address of the local host.

port is the port to which the refused connection was destined.

host is the IP address of the host that would have been refused.

available is the number of connections out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the trace correlator.

probeid is the unique identifier of the probe detection point.

host_current is the number of connections that host held when the connection would have been refused.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9320I TRMD TCP constrained entry logged:*timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **threshold=***threshold*

Explanation: The Intrusion Detection Services (IDS) policy for TCP did not specify TypeActions Limit. The available connections for this port fell to 10% of the value specified in the TotalConnections policy specification.

timestamp is the date and time at which the constrained condition was entered.

lhost is the IP address of the local host.

port is the port that entered constrained state.

host is the IP address of the host causing the constraint.

available is the number of connections available out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

threshold is the constrained state is entered when available is less than or equal to threshold.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9321I TRMD TCP constrained entry:*timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **threshold=***threshold*

Explanation: The Intrusion Detection Services (IDS) policy for TCP specified TypeActions Limit. The available connections for this port fell to 10% of the value specified in the TotalConnections policy specification.

timestamp is the date and time at which the constrained condition was entered.

lhost is the IP address of the local host.

port is the port that entered constrained state.

host is the IP address of the host causing the constraint.

available is the number of connections available out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

threshold indicates that the constrained state is entered when *available* is less than or equal to *threshold*.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9322I TRMD TCP constrained exit logged:*timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **threshold=***threshold*, **duration=***duration*

Explanation: The Intrusion Detection Services (IDS) policy for TCP did not specify TypeActions Limit. The number of available connections for this port exceeded 12% of the value specified in the TotalConnections policy specification.

timestamp is the date and time at which the constrained condition ended.

lhost is the IP address of the local host.

port is the port that is no longer constrained.

host is the IP address of the host causing the constraint.

available is the number of connections available out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

threshold indicates that the constrained state is exited when *available* is greater than *threshold*.

duration is the number of seconds the port was constrained.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9323I TRMD TCP constrained exit:*timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **threshold=***threshold*, **duration=***duration*

Explanation: The Intrusion Detection Services (IDS) policy for TCP specified TypeActions Limit. The number of available connections for this port exceeded 12% of the value specified in the TotalConnections policy specification.

timestamp is the date and time at which the constrained condition ended.

lhost is the IP address of the local host.

port is the port that is no longer constrained.

host is the IP address of the host causing the constraint.

available is the number of connections available out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

threshold indicates that the constrained state is exited when *available* is greater than *threshold*.

duration is the number of seconds the port was constrained.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9324I TRMD TCP connection refused: *timestamp*, **lhost=***lhost*, **port=***port*, **host=***host*, **available=***available*, **total=***total*, **percent=***percent*, **correlator=***correlator*, **probeid=***probeid*, **host_current=***host_current*

Explanation: The Intrusion Detection Services (IDS) policy for TCP specified TypeActions Limit. A connection that did not meet current Traffic Regulation policy specification was refused.

timestamp is the date and time at which the connection was refused.

lhost is the IP address of the local host.

port is the port to which the refused connection was destined.

host is the IP address of the host that was refused.

available is the number of connections available out of the total pool.

total is the number of connections specified by the TotalConnections policy keyword.

percent is the percentage specified by the Percentage policy keyword.

correlator is the IDS trace correlator.

probeid is the unique identifier of the probe detection point.

host_current is the number of connections that host held when the connection was refused.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9325I TRMD Log records missing: *timestamp*, **logtype=***logtype*, **logmissing=***logmissing*

Explanation: The Intrusion Detection Service (IDS) event recording capacity was exceeded and log entries for an intrusion type specified in an active policy have been lost.

timestamp is the date and time at which the log entries were lost.

logtype is the intrusion type for which log entries have been lost. *logtype* will be one of the following:

- ATTACK
- TCPTR
- UDPTR
- SCAN
- SCANDT

logmissing is the number of log entries missing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Examine relevant syslog messages to determine the source of the log entries and either adjust the active policy to be less restrictive or investigate the logged intrusions.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9326I TRMD Statistics records missing: *timestamp*, **stattype=***stattype*, **statmissing=***statmissing*

Explanation: The Intrusion Detection Service (IDS) event recording capacity was exceeded and statistics entries for an intrusion type specified in an active policy have been lost.

timestamp is the date and time at which the statistics entries were lost.

stattype is the intrusion type for which statistics entries have been lost. *stattype* will be one of the following:

- ATTACK
- TCPTR
- UDPTR

statmissing is the number of statistics entries missing.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Examine relevant syslog messages to determine the source of the statistics entries and either adjust the active policy to be less restrictive or investigate the intrusions resulting in statistics logging .

Source Data Set: EZATRMD

Procedure Name: WriteStatEntries

EZZ9327I TRMD Attack log records suppressed: *timestamp*, **attack type=***attacktype*, **count=***count*

Explanation: Intrusion Detection Services (IDS) event recording was suppressed for an attack type specified in the active policy. IDS suppresses logging of attack events of a particular attack type after 100 events have been logged within a 5 minute interval. This is done to prevent syslog flooding. Logging resumes after the 5 minute interval ends.

timestamp is the date and time of the beginning of the 5 minute interval in which log records were suppressed.

attacktype is the attack event type. *attacktype* will be one of the following:

- Malformed
- OutboundRaw
- IPFragment
- ICMP
- IPOPT
- IPPROTO
- Flood
- PerpEcho

These correspond to the AttackType values specified in IDS policy. See the *z/OS Communications Server: IP Configuration Guide* for a description of attack event types.

count is the number of log entries suppressed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Examine relevant syslog messages to determine the source of the log entries and take appropriate action: adjust the active policy to be less restrictive or investigate the logged intrusions.

Source Data Set: EZATRMD

Procedure Name: WriteLogEntries

EZZ9331I get interface configuration: *description*, **errno2=***errnojr*

Explanation: The **named** server failed to get the configuration of a recently created IPv4 interface manager socket.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Ensure that the IPv4 interface the **named** is requesting configuration data for is a valid IPv4 interface.

System Programmer Response: None.

Source Data Set: INTRFCEI

Procedure Name: isc_interfaceiter_create

EZZ9332I get interface configuration: maximum buffer size exceeded

Explanation: The **named** server received more IPv4 interface configuration data than it can accommodate. The data received on the IPv4 interface configuration exceeded 1 megabyte in size.

System Action: The **named** server continues.

User or Operator Response: Ensure that the IPv4 interface the **named** is requesting configuration data for is a valid IPv4 interface.

System Programmer Response: None.

Source Data Set: INTRFCEI

Procedure Name: isc_interfaceiter_create

EZZ9333I *interface_iterator_name*: **getting interface flags:** *description*, **errno2=***errnojr*

Explanation: The **named** server failed to get the network interface flags because of an ioctl error. There is a problem with TCP/IP Services or z/OS. This interface will be ignored.

interface_iterator_name is the interface iterator name.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **IOCTL** option and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Dump the **named** address space if the **named** server did not end. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.
- Dump the TCP/IP Stack. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping the TCP/IP stack.

Source Data Set: INTRFCEI

Procedure Name: internal_current

EZZ9334I *interface_iterator_name*: **getting destination address:** *description*, **errno2=***errnojr*

Explanation: The **named** server failed to get the destination address in a point to point interface because of an ioctl error. There is a problem with TCP/IP Services or z/OS. This interface will be ignored.

interface_iterator_name is the interface iterator name.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **IOCTL** option and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Dump the **named** address space if the **named** server did not end. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.
- Dump the TCP/IP Stack. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping the TCP/IP stack.

Source Data Set: INTRFCEI

Procedure Name: internal_current

EZZ9335I *interface_iterator_name*: **getting netmask:** *description*, **errno2=***errnojr*

Explanation: The **named** server failed to get the network interface network mask because of an ioctl error. There is a problem with TCP/IP Services or z/OS. This interface will be ignored.

interface_iterator_name is the interface iterator name.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **IOCTL** option and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- The component trace requested.
- Dump the **named** address space if the **named** server did not end. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space.
- Dump the TCP/IP Stack. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping the TCP/IP stack.

Source Data Set: INTRFCEI

Procedure Name: internal_current

EZZ9338I could not listen on UDP socket *description*

Explanation: The **named** server failed to support the dispatcher manager or the dispatcher for the UDP socket being listened on because of insufficient memory.

description describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_listenudp

EZZ9339I UDP ns_clientmgr_createclients(): *error_text*

Explanation: The **named** server failed to create a UDP client because of insufficient memory.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_listenudp

EZZ9340I ns_clientmgr_createclients(): *error_text*

Explanation: The **named** server failed to create a client because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: INTRFCEM

Procedure Name: ns_interface_accepttcp

EZZ9341I IPv4: interface iteration failed: *error_text*

Explanation: The **named** server failed to set up an interface iteration because not all interators are complete. This interface will be ignored. This failure can mean that multiple IPv4 interfaces failed. The **named** server might not be listening on all interfaces.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: INTRFCEM

Procedure Name: do_ipv4

EZZ9342I missing SOA

Explanation: The **named** server dynamic update failed to process a resource record (RR) because it could not find a node in the start of authority (SOA) data. The SOA serial number was not updated. The dynamic update action will fail.

System Action: The **named** server continues.

User or Operator Response: Ensure that the zone being updated contains a valid SOA resource record.

System Programmer Response: None.

Source Data Set: JOURNAL

Procedure Name: dns_db_createsoatuple

EZZ9343I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ZONE

Procedure Name: dns_zonemgr_create

EZZ9344I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DNSSDB

Procedure Name: createnode

EZZ9345I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking.

Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: KEYTABLE

Procedure Name: dns_keytable_create

EZZ9346I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to place a read/write lock.

error_text describes the error.

System Action: If the **named** is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: KEYTABLE

Procedure Name: dns_keytable_create

EZZ9347I ns_taskmgr_create() failed: *error_text*

Explanation: The **named** server failed to start the task manager because of insufficient memory.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: NDMAIN

Procedure Name: create_managers

EZZ9348I ns_timermgr_create() failed: *error_text*

Explanation: The **named** server failed to start the timer manager because of insufficient memory.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: NDMAIN

Procedure Name: create_managers

EZZ9349I isc_socketmgr_create() failed: *error_text*

Explanation: The **named** server failed to start the socket manager because of insufficient memory.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: NDMAIN

Procedure Name: create_managers

EZZ9350I isc_entropy_create() failed: *error_text*

Explanation: The **named** server failed to set up the entropy manager because of insufficient memory.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: NDMAIN

Procedure Name: create_managers

EZZ9351I isc_app_run(): *error_text*

Explanation: The **named** server failed to initialize its signal set and returned a result other than reloading or rescanning interfaces.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that TCP/IP Services is active and functioning. Otherwise, contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: NDMAIN

Procedure Name: main

EZZ9352I isc_lex_gettoken() failed: *error_text*

Explanation: The **named** server failed to load a master file because the master file resource record token being processed was not one of the following:

- end-of-line
- end-of-file
- multiple line condition
- escape character

The master file being processed will not be loaded.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the master file being loaded contains valid resource records.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: gettoken

EZZ9353I dns_master_load: *source_file*:*source_line_number*: **$DATE outside epoch**

Explanation: The date specified on the $DATE master file record is incorrect. The master file will not be loaded.

source_file is the source being processed by lex.

source_line_number is the line number in the source file.

System Action: The **named** server continues.

User or Operator Response: Ensure that the date specified on the $DATE master file resource record is correct.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9354I dns_master_load: *source_file*:*source_line_number*: **$DATE in future, using current date**

Explanation: The date specified on the $DATE master file record is incorrect. The current date will be used instead. The master file will be loaded.

source_file is the source being processed by lex.

source_line_number is the line number in the source file.

System Action: The **named** server continues.

User or Operator Response: Ensure that the date specified on the $DATE master file resource record is correct.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9355I *source_file*:*source_line_number*: **isc_lex_gettoken() returned unexpeced token type (***token_type***)**

Explanation: The **named** server failed to process the master file being loaded because it found tokens other than the following:

- end-of-file
- end-of-line
- strings

The master file will not be loaded.

source_file is the source being processed by lex.

source_line_number is the line number lex is processing.

token_type is the token type found in error.

System Action: The **named** server continues.

User or Operator Response: Correct the resource record in error.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9356I isc_lex_gettoken() returned unexpected token type

Explanation: The **named** server failed to process a time-to-live (TTL) resource record in the master file being loaded. The TTL value was not a string. The master file is not loaded.

System Action: The **named** server continues.

User or Operator Response: Ensure that the TTL specified is correct. See the *z/OS Communications Server: IP Configuration Reference* for more information about specifying a TTL value.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9357I isc_lex_gettoken() returned unexpected token type

Explanation: The **named** server failed to process a time-to-live (TTL) resource record in the master file being loaded. The TTL value was not a string. The master file is not loaded.

System Action: The **named** server continues.

User or Operator Response: Ensure that the TTL specified is correct. See the *z/OS Communications Server: IP Configuration Reference* for more information about specifying a TTL value.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9358I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DNSSDB

Procedure Name: dns_sdb_create

EZZ9359I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create a forward table because of a problem creating a read/write lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: FORWARD

Procedure Name: dns_fwdtable_create

EZZ9360I could not set master file style

Explanation: The **named** server failed to convert the resource data set or question section because of a problem with the master file style. The data being processed might be too long or there might be insufficient buffer space to convert the data from column format.

System Action: The **named** server continues.

User or Operator Response: Ensure the master file data being processed is correct. If you are unable to resolve this problem then, contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: MASTERDU

Procedure Name: dns_rdataset_totext

EZZ9361I master file write failed: *error_text*

Explanation: The **named** server failed to write the domain resource data to a master file.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the **named** server has the authority to write to a master file. Use the error response to determine why the **named** server could not write the buffer to the master file.

System Programmer Response:

Source Data Set: MASTERDU

Procedure Name: dump_rdataset

EZZ9362I could not set master file style

Explanation: The **named** server failed to convert the resource data set or question section because of a problem with the master file style. The data being processed might be too long or there might be insufficient buffer space to convert the data from column format.

System Action: The **named** server continues.

User or Operator Response: Ensure the master file data being processed is correct. If you are unable to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: MASTERDU

Procedure Name: dns_master_dumptostream

EZZ9363I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking.

Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ISCMEM

Procedure Name: isc_mem_createx

EZZ9364I socket() failed: *errno_text*, **errno2=***errnojr*

Explanation: The **named** server failed to create a socket.

errno_text describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Use the error response to determine why the **named** server could not create a socket. If you are not able to resolve the problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: NET

Procedure Name: try_proto

EZZ9365I unknown state in omapi_protocol_signal_handler: *protocol_state*

Explanation: The **named** server object manager failed to process a protocol signal because of an incorrect protocol state. The **named** object manager wakes up after a number of bytes were processed to check for signals.

protocol_state is the state of the signal handler. Valid protocol states are as follows:

- intro wait
- header wait
- name length wait
- name wait
- value length wait
- value wait
- signature wait

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: OMAPPROT

Procedure Name: dispatch_messages

EZZ9366I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: RBTDBJJM

Procedure Name: dns_rbtdb_create

EZZ9367I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create the DNS Red Black Tree (RBT) database read/write lock because of insufficient memory.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: RBTDBJJM

Procedure Name: dns_rbtdb_create

EZZ9368I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: RBTDBJJM

Procedure Name: dns_rbtdb_create

EZZ9369I isc_lex_gettoken() failed: *error_text*

Explanation: The **named** server failed to convert resource data from text format. Processing the next resource data token failed because the last token processed returned an error condition other than insufficient buffer space or insufficient memory.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: None.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: RDATA

Procedure Name: dns_rdata_fromtext

EZZ9370I unexpected non-minimal diff

Explanation: The **named** server is determining if there is a difference in resource records while performing a dynamic update through an add, delete or assert of an resource record. A minimal compare is made to check for for duplicate owner names, time-to-live, and resource data.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DNSDIFF

Procedure Name: dn_diff_appendminimal

EZZ9371I isc_timer_reset(): *error_text*

Explanation: The **named** server failed to get the time of day while resetting the resolver timer to an inactive state. The time returned might not be accurate.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPUs date is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: RESOLVER

Procedure Name: fctx_stoptimer

EZZ9372I isc_time_nowplusinterval: *error_text*

Explanation: The **named** server failed to compute the resolver fetch expiration time.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: RESOLVER

Procedure Name: fctx_create

EZZ9373I isc_timer_create: *error_text*

Explanation: The **named** server failed to create the resolver fetch inactivity timer.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPUs date is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: RESOLVER

Procedure Name: fctx_create

EZZ9374I isc_result_register() failed: *error_code*

Explanation: The **named** server failed to register an internal results table used to report function results in textual format because of insufficient memory.

*error_code*describes the error and will be a 1 for insufficient memory.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: DNSRESUL

Procedure Name: initialize_action

EZZ9375I register_table() failed: *error_code*

Explanation: The **named** server failed to register an internal results table used to report function results in textual format because of insufficient memory.

error_code describes the error and will be a 1 for insufficient memory.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: ISCRESUL

Procedure Name: initialize_action

EZZ9376I isc_result_register() failed: *error_code*

Explanation: The **named** server failed to register an internal results table used to report function results in textual format because of insufficient memory.

error_code describes the error and will be a 1 for insufficient memory.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: RESULTCC

Procedure Name: initialize_action

EZZ9377I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server might end depending on the severity of the problem.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: RWLOCK

Procedure Name: print_lock

EZZ9378I isc_condition_init(readable) failed: *error_text*

Explanation: The **named** server failed to initialize the thread condition variable because of insufficient memory. The read/write lock conditional variable is not set.

error_text describes the error.

System Action: The **named** server might end depending on the severity of the problem.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: RWLOCK

Procedure Name: print_lock

EZZ9379I isc_condition_init(writeable) failed: *error_text*

Explanation: The **named** server failed to initialize the thread condition variable because of insufficient memory. The read/write lock conditional variable is not set.

error_text describes the error.

System Action: The **named** server might end depending on the severity of the problem.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: RWLOCK

Procedure Name: print_lock

EZZ9380I unknown address family: *socket_address_family*

Explanation: The **named** found an incorrect socket address family for the socket address being hashed. The valid socket address families are AF_INET or AF_INET6 addresses. The hash value will not be provided for the socket address.

socket_address_family is the socket address family found in error.

System Action: The **named** server continues.

User or Operator Response: Ensure that all socket addresses are AF_INET or AF_INET6 addresses.

System Programmer Response: None.

Source Data Set: SOCKADDR

Procedure Name: isc_sockaddr_hash

EZZ9381I fcntl(*file_descriptor***, F_SETFL,** *flag_set***):** *error_text***, errno2=***errnojr*

Explanation: The **named** server failed to change a socket to a non-blocking status.

file_descriptor is the file_descriptor the fcntl() is using.

flag_set is the set of flags that fcntl() is using.

error_text describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: If the **named** server was creating a socket manager, it ends. Otherwise, it continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the

TC, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: make_nonblock

EZZ9382I internal_send: *address*: *error_text*

Explanation: The **named** server failed to send a message to a socket.

address is the ip address that is failing an internal send.

error_text describes the error.

System Action: The **named** server continues

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: doio_send

EZZ9383I internal_send: send() returned 0

Explanation: The **named** server sent zero bytes of data to the client. This is not an error.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ISCSOCKT

Procedure Name: doio_send

EZZ9384I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ISCSOCKT

Procedure Name: allocate_socket

EZZ9385I socket() failed: *error_text*, **errno2=***errnojr*

Explanation: The **named** server failed to create a socket.

error_text describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_create

EZZ9390I internal_accept: accept() failed: *error_text*, **errno2=***errnojr*

Explanation: The **named** server accept() failed.

error_text describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Ensure that TCP/IP Services is available. Otherwise, contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: internal_accept

EZZ9392I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ISCSOCKT

Procedure Name: isc_socketmgr_create

EZZ9393I isc_condition_init() failed

Explanation: An error condition occurred while setting the shutdown_ok condition for the socket manager being created because of insufficient memory. The socket manager is not created.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: ISCSOCKT

Procedure Name: isc_socketmgr_create

EZZ9394I pipe() failed: *error_text*, **errno2=***errnojr*

Explanation: The **named** server failed to create a pipe.

error_text describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socketmgr_create

EZZ9395I isc_thread_create() failed

Explanation: The **named** server failed to start the select/pool thread because of insufficient memory or the system cannot support another thread.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server. Ensure that the user ID used by the **named** server has the correct security privileges. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ISCSOCKT

Procedure Name: isc_socketmgr_create

EZZ9396I isc_thread_join() failed

Explanation: The **named** server failed to destroy the socket manager because the thread will not end. The thread might be incorrect, undetached, or in a deadlock condition. Thread clean up continues.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socketmgr_destroy

EZZ9397I setsockopt(*socket_file_descriptor***) failed**

Explanation: An attempt to set socket option SO_REUSEADDR in isc_socket_bind() or SO_EioIfNewTP in isc_socket_create() failed. Socket bind continues.

socket_file_descriptor

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_bind/isc_socket_create

EZZ9398I bind: *error_text*, **errno=***errno*, **errno2=***errnojr*

Explanation: The **named** server bind() failed.

error_text describes the error.

errno is the decimal Return Code. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_bind

EZZ9399I listen: *description*, **errno2=***errnojr*

Explanation: The **named** server listen() failed.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_listen

EZZ9400I *description/errno*, **errno2=***errnojr*

Explanation: The **named** server failed to connect a socket.

description describes the error.

errno is the decimal Return Code. Return Codes are listed and described in the *z/OS UNIX System Services Messages and Codes*.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_connect

EZZ9401I internal_connect: connect() *description*, **errno2=***errnojr*

Explanation: The **named** server encountered a socket error while processing a pending socket connection.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the

TC, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: internal_connect

EZZ9402I getsockname: *description*, **errno2=***errnojr*

Explanation: The **named** server failed to get a socket name for a connection. The connection ends.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_getsockname

EZZ9403I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TASK

Procedure Name: isc_task_create

EZZ9404I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TASK

Procedure Name: isc_taskmgr_create

EZZ9405I isc_condition_init() failed

Explanation: The **named** server failed to initialize a conditional variable because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: TASK

Procedure Name: isc_taskmgr_create

EZZ9406I *description*, **errno2=***errnojr*

Explanation: The **named** server failed to obtain the time of day. Messages will not have the correct time values.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: ISCTIME

Procedure Name: isc_time_now

EZZ9407I *description*, **errno2=***errnojr*

Explanation: The **named** server failed to obtain the time of day. Messages will not have the correct time values.

description describes the error.

errnojr is the hexadecimal Reason Code. The format of the four byte reason code is explained in the introduction to the Reason Code section of the *z/OS UNIX System Services Messages and Codes*.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: ISCTIME

Procedure Name: isc_time_nowplusinterval

EZZ9408I isc_time_now() failed: *error_text*

Explanation: The **named** server failed to create a timer while obtaining the time of day. An internal manager will not have a timer.

error_text describes the error.

System Action: If the **named** server was processing server functions it ends. If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: TIMER

Procedure Name: isc_timer_create

EZZ9409I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TIMER

Procedure Name: isc_timer_create

EZZ9410I isc_time_now() failed: *error_text*

Explanation: The **named** server failed to obtain the time of day. The timers type, expire, and interval value will not be changed. the current time.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: TIMER

Procedure Name: isc_timer_reset

EZZ9411I isc_time_now() failed: *error_text*

Explanation: The **named** server failed to obtain the time of day. The timer will not have the currect time.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command.

System Programmer Response: None.

Source Data Set: TIMER

Procedure Name: isc_timer_touch

EZZ9412I couldn't allocate event

Explanation: The **named** server timer dispatcher failed to post an event because of insufficient memory. A thread will not be dispatched.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: TIMER

Procedure Name: dispatch

EZZ9413I couldn't schedule timer: *result*

Explanation: The **named** server failed to schedule a thread timer because of insufficient memory or the time of day is incorrect. A thread will not be dispatched. The event is not scheduled.

result is the result returned from the schedule function.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use the z/OS UNIX **date** command to ensure that the CPU time is correct. See the *z/OS UNIX System Services Command Reference* for more information about the **date** command. Otherwise, increase the TSO address space or region size and restart the **named** server.

Source Data Set: TIMER

Procedure Name: dispatch

EZZ9414I isc_mutex_init() failed

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TIMER

Procedure Name: isc_timermgr_create

EZZ9415I isc_condition_init() failed

Explanation: The **named** server failed to create a condition variable for a thread because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: TIMER

Procedure Name: isc_timermgr_create

EZZ9416I isc_thread_create() failed

Explanation: The **named** server failed to create a thread because of one of the following reasons:

- The system could not create another thread.
- The thread is not valid.
- There is not enough memory to create the thread.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure adequate resources are available for the thread or increase the TSO address space or region size and restart the **named** server. Otherwise, re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.

- The component trace requested.
- A dump of the **named** server address space. See the *z/OS Communications Server: IP Diagnosis* for information about dumping an address space.

Source Data Set: TIMER

Procedure Name: isc_timermgr_create

EZZ9417I isc_thread_join() failed

Explanation: The **named** server failed waiting for a thread to end. The **named** server is shutting down and continues with the clean up process.

System Action: The **named** server ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: TIMER

Procedure Name: isc_timermgr_destroy

EZZ9419I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create a transaction signature lock because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: TSIG

Procedure Name: dns_tsigkeyring_create

EZZ9420I temp entry creation failed: *error_text*

Explanation: The **named** server failed to temporarily append a resource record (RR) set because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: UPDATE

Procedure Name: update_action

EZZ9421I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking.

Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: VIEW

Procedure Name: dns_view_create

EZZ9422I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create a view lock because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: VIEW

Procedure Name: dns_view_create

EZZ9423I dns_zt_create() failed: *error_text*

Explanation: The **named** server failed because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server. Ensure that the user ID used by the **named** server has the correct security privileges. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: VIEW

Procedure Name: dns_view_create

EZZ9424I dns_keytable_create() failed: *error_text*

Explanation: The **named** server failed to create a key table for secured roots because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server. Ensure that the user ID used by the **named** server has the correct security privileges. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: VIEW

Procedure Name: dns_view_create

EZZ9425I dns_keytable_create() failed: *error_text*

Explanation: The **named** server failed to create a key table for trusted keys because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Ensure that the user ID used by the **named** server has the correct security privileges. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: VIEW

Procedure Name: dns_view_create

EZZ9426I isc_mutex_init() failed: *error_text*

Explanation: An attempt to aquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ZONE

Procedure Name: dns_zone_create

EZZ9427I unexpected zone type *zone_type*

Explanation: The **named** server found a zone with a type other than master, slave, or stub. The zone will not be loaded or reloaded.

zone_type is the zone type of the zone being processed.

System Action: The **named** server continues.

User or Operator Response: Ensure that the zone type being processed is master, slave, or stub.

System Programmer Response: None.

Source Data Set: ZONE

Procedure Name: dns_zone_load

EZZ9428I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create a zone manager lock because of insufficient memory.

error_text describes the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ZONE

Procedure Name: dns_zonemgr_create

EZZ9429I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create a zone manager configuration data read/write lock because of insufficient memory.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ZONE

Procedure Name: dns_zonemgr_create

EZZ9430I isc_rwlock_init() failed: *error_text*

Explanation: The **named** server failed to create a zone table read/write lock because of insufficient memory. The **named** server is creating a view or a bind view, or searching for a view.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the user ID used by the **named** server has sufficient authority to perform locking. Increase the TSO address space or region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: ZT

Procedure Name: dns_zt_create

EZZ9500I forward zones do not have a max-retry-time field

Explanation: The max-retry-time option is specified in a forward zone in the **named** server configuration file. The option max-retry-time is only allowed in master, slave, or stub zones. This option is ignored.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the max-retry-time option from all forward zones or hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxretrytime

EZZ9501I hint zones do not have a min-refresh-time field

Explanation: The min-refresh-time option is specified in a hint zone in the **named** server configuration file. The option min-refresh-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the min-refresh-time option from all forward zones or hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setminrefreshtime

EZZ9502I forward zones do not have a min-refresh-time field

Explanation: The min-refresh-time option is specified in a forward zone in the **named** server configuration file. The option min-refresh-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the min-refresh-time option from all forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setminrefreshtime

EZZ9503I hint zones do not have a min-refresh-time field

Explanation: The min-refresh-time option is specified in a hint zone in the **named** server configuration file. The option min-refresh-time is only allowed in master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the min-refresh-time option from all hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getminrefreshtime

EZZ9504I forward zones do not have a min-refresh-time field

Explanation: The min-refresh-time option is specified in a forward zone in the **named** server configuration file. The option min-refresh-time is only allowed in master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the min-refresh-time option from all forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getminrefreshtime

EZZ9505I hint zones do not have a max-refresh-time field

Explanation: The max-refresh-time option is specified in a hint zone in the **named** server configuration file. The option max-refresh-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-refresh-time option from all hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxrefreshtime

EZZ9506I forward zones do not have a max-refresh-time field

Explanation: The max-refresh-time option is specified in a forward zone in the **named** server configuration file. The option max-refresh-time is only allowed in master, slave, or stub zones.

System Action: The **named** server ends.

User or Operator Response: Remove the max-refresh-time option from all forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_setmaxrefreshtime

EZZ9507I hint zones do not have a max-refresh-time field

Explanation: The max-refresh-time option is specified in a hint zone in the **named** server configuration file. The option max-refresh-time is only allowed in master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the max-refresh-time option from all hint zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxrefreshtime

EZZ9508I forward zones do not have a max-refresh-time field

Explanation: The max-refresh-time option is specified in a forward zone in the **named** server configuration file. The option max-refresh-time is only allowed in master, slave, or stub zones.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Remove the max-refresh-time option from all forward zones.

System Programmer Response: None.

Source Data Set: CONFZONE

Procedure Name: dns_c_zone_getmaxrefreshtime

EZZ9509I unsupported database type *database_type*

Explanation: The **named** server failed to build a database because the database requested is not supported.

database_type is the type of database.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system Programmer

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Dump the **named** address space. See the *z/OS Communications Server: IP Diagnosis* for instructions on dumping an address space. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use the **rndc dumpdb** utility to dump the **named** servers cache to a file.
- The component trace requested.
- The dump of the **named** address space.

Source Data Set: DB

Procedure Name: dns_db_create

EZZ9510I dumping master file: *file_name***: close:** *error_text*

Explanation: The **named** server failed to close the dump file.

file_name is the dump file name.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that there is enough disk space to hold the master file being dumped.

System Programmer Response: None.

Source Data Set: MSTERDU

Procedure Name: dns_master_dumpnode

EZZ9511I $GENERATE: *source_name*:*source_line*: **invalid range** *range*

Explanation: The **named** server does not load the zone data. An invalid range is specified as a parameter to the zone file directive $GENERATE. See the *z/OS Communications Server: IP Configuration Reference* for details on the correct syntax of the $GENERATE directive.

source_name is the file name of a zone file that contains a $GENERATE error.

source_line is the zone file line number that contains a $GENERATE error.

range is the range parameter of the $GENERATE directive.

System Action: The **named** server continues.

User or Operator Response: Specify a valid range to the $GENERATE directive and restart the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: generate

EZZ9512I $GENERATE: *source_name*:*source_line*: **unknown RR type** *gtype*

Explanation: The **named** server does not load the zone data. An unknown resource record type is specified as a parameter to the zone file directive $GENERATE. See the *z/OS Communications Server: IP Configuration Reference* for details on the correct syntax of the $GENERATE directive.

source_name is the file name of a zone file that contains a $GENERATE error.

source_line is the zone file line number that contains a $GENERATE error.

gtype is the resource record type that is not supported by the $GENERATE directive.

System Action: The **named** server continues.

User or Operator Response: Specify a valid RR type to the $GENERATE directive and restart the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: generate

EZZ9513I $GENERATE: *source_name*:*source_line*: **unsupported type** *gtype*

Explanation: The **named** server does not load the zone data. An unsupported resource record type is specified as a parameter to the zone file directive $GENERATE. The supported resource record (RR) types for the $GENERATE directive are name server (NS), pointer (PTR), conical name (CNAME), address (A) and IPv6 address (AAAA). See the *z/OS Communications Server: IP Configuration Reference* for details on the correct syntax of the $GENERATE directive.

source_name is the file name of a zone file that contains a $GENERATE error.

source_line is the zone file line number that contains a $GENERATE error.

gtype is the resource record type that is not supported by the $GENERATE directive.

System Action: The **named** server continues.

User or Operator Response: Specify a valid RR type to the $GENERATE directive and restart the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: generate

EZZ9514I $GENERATE: *error_text*

Explanation: An out of memory error was encountered while processing a $GENERATE zone file directive. The **named** server does not load the zone data.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: MASTER

Procedure Name: generate

EZZ9515I $GENERATE: *source_name*:*source_line*: *error_text*

Explanation: An error other an insufficient memory occured while processing a $GENERATE direcive. The **named** server does not load the zone data.

source_name is a file name of a zone file that contains a $GENERATE error.

source_line is the zone file line number that contains the $GENERATE error.

error_text is the error describing the problem with the $GENERATE directive.

System Action: The **named** server continues.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use the **rndc dumpdb** utility to dump the **named** servers cache to a file.
- The component trace requested.

Source Data Set: MASTER

Procedure Name: generate

EZZ9516I dns_master_load: *source_name*:*source_line*: **no TTL specified**

Explanation: No $TTL directive was specified in the *source_name* zone file and no start of authority (SOA) resource record (RR) time-to-live (TTL) is specified to act as a default TTL value. In this case, the $TTL directive is required to provide a default value. Consult RFC 1035 and RFC 2308 for more details on zone file TTL values. The **named** server does not load the zone data. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for directions to get a copy of the RFC.

source_name is the file name of the zone file that lacks a $TTL directive.

source_line is the line number of the first resource record that does not have an explicit TTL specified.

System Action: The **named** server continues.

User or Operator Response: Specify a $TTL value or specify an SOA record TTL value to act as the default TTL.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9517I dns_master_load: *source_name*:*source_line*: *error_text*

Explanation: An error other than insufficient memory occurred while processing a zone file. The **named** server does not load the zone data.

source_name is the zone file name.

source_line is the zone file line number that caused the error.

error_text describes the error.

System Action: The **named** server might end depending on the severity of the problem.

User or Operator Response: Correct the problem described by this message. Otherwise, contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file.
- The SYSLOG DAEMON logging file.
- All configured **named** logs.
- Use the **rndc dumpdb** utility to dump the **named** servers cache to a file.
- The component trace requested.

Source Data Set: MASTER

Procedure Name: load

EZZ9518I lame server on *name* **(in** *domain***?):** *address*

Explanation: The **named** server found a server that is registered as authoritative for some zone but is not itself configured as authoritative for that zone. Lame servers are misconfigurations in remote servers, discovered by BIND 9 when trying to query those servers during resolution.

name is the name of lame server.

domain is the domain found to be associated with the host.

address is the address of the host.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: RESOLVER

Procedure Name: log_lame

EZZ9519I the TSIG key for *name* **is too short to be secure**

Explanation: The **named** configuration file defines a transaction signature (TSIG) key that is fewer than 64 bits in length. Keys smaller than 64 bits are insecure and should not be used.

name is the name of the TSIG key.

System Action: The **named** server continues.

User or Operator Response: Ensure that the keys used is at least 64 bits in length.

System Programmer Response: None.

Source Data Set: TSIG

Procedure Name: dns_tsigkey_createfromkey

EZZ9520I TTL differs in rdataset, adjusting *TTL_data* **->** *rdatalist_TTL*

Explanation: The **named** server set the resource record being added time-to-live (TTL) value to that specified by a previous resource record.

TTL_data is the TTL for the resource record being added.

rdatalist_TTL is the TTL value that is being used for the new resource record.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DNSDIFF

Procedure Name: dns_diff_apply

EZZ9521I update with no effect

Explanation: The **named** server encountered an update that did not produce any changes to the zone data files. This is not an error.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DNSDIFF

Procedure Name: dns_diff_apply

EZZ9522I update with no effect

Explanation: The **named** server encountered an update that did not produce any changes to the zone data files. This is not an error.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DNSDIFF

Procedure Name: dns_diff_load

EZZ9523I dns_fwdtable_create() failed: *error_text*

Explanation: The **named** server failed to create a forwarding table because of insufficient memory.

error_text describes the error.

System Action: If the **named** server is initially loading, and it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the TSO address space or region size and restart the **named** server.

Source Data Set: VIEW

Procedure Name: dns_view_create

EZZ9534I secure zone with no NXTs

Explanation: The **named** server encountered a DNSSEC-secured zone with no next (NXT) records while attempting to dynamically update the zone.

System Action: The **named** server continues.

User or Operator Response: Shut down the **named** server with **rndc stop** utility to get an up to date zone file for the dynamic zone, re-sign the whole zone using the **dnssec-signzone** utility, remove the zones journal, and restart the **named** server.

System Programmer Response: None.

Source Data Set: UPDATE

Procedure Name: next_active

EZZ9535I configuring trusted key for *trusted_key_domain*: *error_text*: *file_name*:*line_number*

Explanation: There was an error setting up a trusted key that was specified in the **named** server configuration file for the *trusted_key_domain* domain.

trusted_key_domain is the domain of the trusted key being configured.

error_text describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the trusted key specified is correct. Trusted keys can be generated by using the **dnssec-keygen** utility. See the *z/OS Communications Server: IP System Administrator's Commands* for information about using the **dnssec-keygen** utility to generate trusted keys.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_view_dnsseckey

EZZ9538I query logging is now *on_off_indicator*

Explanation: The **named** server is reporting the status of the query logging feature of the logging facility. The **named** server will record all queries as indicated. *on_off_indicator* is used to indicate whether query logging is on or off.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: ns_server_togglequerylog

EZZ9539I BIND *version*

Explanation: The *version* reports the BIND DNS server level on which the CS/390 **named** server is based.

version is the version of DNS/Bind reported from the -v command line option.

System Action: The **named** server ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: parse_command_line

EZZ9540I using *number_of_cpus* **CPU***singular_or_plural*

Explanation: The **named** server detected and is utilizing the specified number of CPUs on the host system.

number_of_cpus is the number of CPUs the **named** server will use.

singular_or_plural is null for one CPU and the letter s for more than one CPU.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: create_managers

EZZ9541I configuring key *key_id*: *error_text*: *file_name*:*line_number*

Explanation: The **named** server encountered an error processing a transaction signature (TSIG) key.

key_id is a key name specified in the **named** server configuration file.

error_text is the error returned while trying to add additional keys.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: If the **named** server is initially loading, it stops. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Ensure that the key is correct. See the *z/OS Communications Server: IP Configuration Reference* for information about specifying a key statement in the **named** server configuration file for use in securing transactions.

System Programmer Response: None.

Source Data Set: TSIGCONF

Procedure Name: add_initial_keys

EZZ9542I zone *zone_name* **allows updates by IP address, which is insecure**

Explanation: The specifed zone allows the authorization for dynamic updating of zone data by IP address. Authentication by IP address is susceptible to spoofing. Authorization based on transaction signature (TSIG) keys is recommended.

zone_name is a zone name specified in the **named** server configuration file.

System Action: The **named** server continues.

User or Operator Response: Specify a TSIG key name in the allow-update clause of the zone statement to suppress the warning message and ensure dynamic update security.

System Programmer Response: None.

Source Data Set: ZONECONF

Procedure Name: ns_zone_configure

EZZ9543I notify failed: *failing_reason error_text*

Explanation: The **named** server failed to process a notify request.

failing_reason describes the error.

error_text is the error code in text from the zone change notification process.

System Action: The **named** server ignores the notify message and continues.

User or Operator Response: Set the debug level to three in the **named** logging files to determine the source of the the notify requested.

System Programmer Response: None.

Source Data Set: NOTIFY

Procedure Name: FAILC

EZZ9547I starting named, BIND *version command_line_arguments*

Explanation: The **named** server begins loading using the command line arguments specified.

version is the version of DNS/Bind being started.

command_line_arguments are the command line arguments that are being used to start the **named** server.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ9548I named BIND 9 is using default messages

Explanation: The **named** server will use the English only version of its messages.

System Action: The **named** server continues.

User or Operator Response: Ensure a valid message catalog file named ns9.cat exists in the path specified by the NLSPATH z/OS UNIX shell environment variable.

System Programmer Response: None.

Source Data Set: B9MSGCAT

Procedure Name: ibm_msgcat_init

EZZ9549I named BIND 9 is using message catalog *msgcat_name*

Explanation: The **named** server will be retrieving its messages from the *msgcat_name* message catalog.

msgcat_name is the name of the message catalog where message are extracted.

System Action: The **named** server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: B9MSGCAT

Procedure Name: ibm_msgcat_init

EZZ9550I UDP client handler shutting down due to fatal receive error: *error_text*, *result_text*

Explanation: An unrecoverable error was encountered while receiving a UDP request. *error_text* and *result_text* describe the error.

System Action: The related **named** server request will not be answered. The **named** server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the PFS, SOCKET and UDP options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: CLIENT

Procedure Name: client_request

EZZ9551I port *port_number* **out of range:** *file_name*:*line_number*

Explanation: The port number specified by the port clause on the **also-notify** zone statement in named.conf is greater than 65535.

port_number is the value that is out of range.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: Specify a port number that is within the valid range in the configuration file, and restart the **named** server.

System Programmer Response: None.

Source Data Set: CONFIG

Procedure Name: ns_config_getiplist

EZZ9552I port *port_number* **out of range:** *file_name*:*line_number*

Explanation: The port number specified by the port clause on the 'masters' zone statement in file_name is greater than 65535.

port_number is the value that is out of range.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: Specify a port number that is within the valid range in the configuration file, and restart the **named** server.

System Programmer Response: None.

Source Data Set: CONFIG

Procedure Name: ns_config_getipandkeylist

EZZ9553I port *port_number* **out of range:** *file_name*:*line_number*

Explanation: The port number, port_number, specified by a port clause in named.conf is greater than 65535.

port_number is the value that is out of range.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends if it was starting. The **named** server continues if it was reloading.

User or Operator Response: Change the value for the **port** option to a valid value and restart the **named** server.

System Programmer Response: None.

Source Data Set: CONFIG

Procedure Name: ns_config_getport

EZZ9554I unknown control channel command *command*

Explanation: An unrecognized rndc command, command, was received by the **named** server.

command is the unrecognized rndc command received.

System Action: The unrecognized rndc command is ignored. The **named** server continues.

User or Operator Response: Issue the rndc command without any options or see the rndc man page for a valid list of rndc commands.

System Programmer Response: None.

Source Data Set: CONTROL

Procedure Name: ns_control_docommand

EZZ9556I couldn't register key *key_id***: out of memory:** *file_name***:***line_number*

Explanation: There was a problem registering the *key_id* because of insufficient memory . The control facility will be unavailable for use with the key *key_id*.

key_id is the name of the key being registered.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer to increase memory.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: CNTLCONF

Procedure Name: register_keys

EZZ9557I error: *error_text* **- during zone transfer of the following RR:** *resource_record*

Explanation: An error was detected when receiving a zone transfer. A resource record (RR) was being processed at the time the error was detected.

error_text explains the error.

resource_record is the resource record that was being processed at the time the error was detected. The DNS name in the resource record is fully qualified, thus showing the zone name affected by the error.

System Action: The zone in which the error was detected is not loaded by this name server. The name server continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Contact the DNS administrator of the name server the zone was being transferred from and inform them of this error. The zone name is contained within the DNS name of *resource_record*. Usually this error is caused by a back-level name server that allows illegal resource records or combinations of resource records. The errors should be corrected on the primary name server of the zone in error.

Source Data Set: dns_diff_load

Procedure Name: DNSDIFF

EZZ9558I stopping command channel on *socket_address*

Explanation: The **named** server socket on *socket_address* is being closed and will no longer be used to process commands from an rndc client. rndc will no longer communicate with this **named** server on the socket, *socket_address*.

socket_address is the IP address on which rndc is no longer listening.

System Action: The **named** server continues. The control channel was most likely stopped by a server shutdown or

control channel reconfiguration. rndc client commands will no longer be processed by this **named** server until a new control channel is opened.

User or Operator Response: If the command channel was not closed intentionally, check the logs for previous messages describing why the channel was closed.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: shutdown_listener

EZZ9559I error sending command response to *socket_address*: *error_text*

Explanation: The **named** server was attempting to send a response to the rndc client over *socket_address* but was unable to do so. A possible reason is that the control channel for the rndc client might have been closed.

socket_address is the IP address over which rndc encountered an error.

error_text gives more information about the error.

System Action: The **named** server continues and the command response is not retried.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the PFS, SOCKET and TCP options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: CNTLCONF

Procedure Name: control_senddone

EZZ9560I invalid command from *socket_address*: *error_text*

Explanation: An unrecognized rndc command or an rndc command with improper syntax was received by the **named** server on *socket_address*.

socket_address is the IP address over which rndc encountered an error.

error_text contains more information about the error.

System Action: The **named** server continues. The unrecognized rndc command is ignored.

User or Operator Response: Check the product documentation, the rndc man page, or the rndc help text for the supported rndc commands and syntax.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: log_invalid

EZZ9561I rejected command channel message from *socket_address*

Explanation: The rndc command cannot be processed because the rndc client at *socket_address* does not have the necessary authority to send commands to this **named** server. The address match list on the **allow** phrase of the **controls** statement in named.conf does not include the client at *socket_address*.

socket_address is the IP address over which rndc encountered an error.

System Action: The rndc command is not processed. The **named** server continues.

User or Operator Response: Issue the rndc command from a host that the rndc command is allowed by the **named** server. If this rndc client should have authority, add *socket_address* to the **allow** phrase of the **controls** statement in

named.conf. The rndc client may be multi-homed and possibly the rndc client is using a different source address than the one expected.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: control_newconn

EZZ9562I dropped command channel from *socket_address*: *result_text*

Explanation: The **named** server was unable to read data on the rndc control channel TCP connection from the rndc client at *socket_address*, most likely because the TCP connection between the rndc client and **named** server was dropped.

socket_address is the IP address over which rndc encountered an error.

result_text contains more information about the error.

System Action: The rndc command is not processed. The **named** server continues.

User or Operator Response: Try the rndc command again. If unsuccessful, contact the system programmer.

System Programmer Response: Use the netstat command to verify that the **named** server has an open socket on the expected address and port. If not, modify the **controls** statement in named.conf and reload the **named** server. Verify that the rndc client is attempting to contact the same address and port. This might involve checking the rndc.conf file, named.key file, and the rndc program options used by the client. If the problem still cannot be resolved, re-create the problem after setting the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The rndc.conf, named.key, and rndc command syntax used by the rndc client

Source Data Set: CNTLCONF

Procedure Name: control_newconn

EZZ9563I unsupported algorithm *alg_name* **in key** *key_id* **for use with command channel** *socket_addr* **:***file_name***:***line_number*

Explanation: The **named** server control facility only supports keys generated with the hmac-md5 algorithm. The control facility (rndc functions) will be unavailable for use with the key *key_id*.

alg_name is the name of the algorithm specified to interpret the key registration.

key_id is the ID of the key being registered.

socket_addr is the socket IP address for the object manager being configured.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues. rndc functions might be unavailable for this **named** server.

User or Operator Response: Ensure that the key *key_id* is using algorithm hmac-md5. See the *z/OS Communications Server: IP Configuration Reference* for instructions for supplying an algorithm with a key in the **named** server configuration file on the **key** statement.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: get_rndckey

EZZ9564I secret for key *key_name* **on command channel:** *error_text*: *file_name*:*line_number*

Explanation: The **named** server failed to decode the key *key_name*. The specified key might have been created incorrectly. The key's secret must be a valid base64 encoded string. The valid base64 characters are:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=
```

The control facility will be unavailable for use with the key *key_name*.

key_name is the name of the key being decoded.

error_text describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Ensure that the key is correctly generated. See *z/OS Communications Server: IP System Administrator's Commands* for information about using the dnssec-keygen utility for generating keys.

System Programmer Response: None.

Source Data Set: CNTLCONF

Procedure Name: get_rndckey

EZZ9565I couldn't register key *key_id*: **out of memory:** *file_name*:*line_number*

Explanation: There was a problem registering the *key_id* because of insufficient memory . The control facility will be unavailable for use with the key, *key_id*.

key_id is the name of the key being registered.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues.

User or Operator Response: Contact the system programmer to increase memory.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: CNTLCONF

Procedure Name: get_rndckey

EZZ9566I ignoring trusted key for '*key_name***': no crypto support:** *file_name*:*line_number*

Explanation: The **named** server was compiled without support for OpenSSL. The trusted key will be ignored and consequently, DNSSEC authentication will not occur.

key_name is the name of the key being processed when the error occurred.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: Notify the system programmer.

System Programmer Response: This is an internal error. Perform a dump of the **named** server address space, and contact the IBM Software Support Center with the following additional documentation:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs

Source Data Set: SERVER

Procedure Name: configure_view_dnsseckey

EZZ9567I no more recursive clients: *error_text*

Explanation: The **named** server is temporarily unable to allocate the necessary resources to accept new queries because of a large amount of outstanding recursive queries. When the answers to the outstanding queries are returned to this **named** server, the resource shortage will be relieved.

System Action: The **named** server continues. New queries to the **named** server might temporarily be refused.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: QUERY

Procedure Name: query_recurse

EZZ9568I couldn't install new keys for command channel *socket_address*: *error_text*: *file_name*:*line_number*

Explanation: The rndc control channel is being updated because of changes in the **controls** statement after the **named** server was reloaded. There was insufficient memory to process the related keys. The new keys configured in the **controls** statement will be unavailable for use with the control facility.

socket_address is the socket IP address for the control channel being configured.

error_text describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues. The new keys configured in the **controls** statement will be unavailable for use with the control facility.

User or Operator Response: Contact the system programmer to increase memory.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: CNTLCONF

Procedure Name: update_listener

EZZ9569I couldn't install keys for command channel *socket_address*: *error_text*: *file_name* :*line_number*

Explanation: An rndc control channel is being added because of a reload of the **named** server after additional **controls** statements were added to named.conf. There was insufficient memory to process the related keys. The keys configured in the **controls** statement will be unavailable for use with the control facility.

socket_address is the socket IP address for the control channel being configured.

error_text describes the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server continues. The keys configured in the added **controls** statement will be unavailable for use with the control facility.

User or Operator Response: Contact the system programmer to increase memory.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: CNTLCONF

Procedure Name: add_listener

EZZ9570I 'additional-from-auth no' is only supported with 'recursion no'

Explanation: A value of **no** for the **additional-from-auth** option is incompatible when recursion is disabled.

System Action: The **named** server continues.

User or Operator Response: The **additional-from-auth** option should be enabled in the named.conf file, or recursion should be enabled with the **recursion** option in named.conf. After the changes have been made, reload the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_view

EZZ9571I program name too long

Explanation: The internal name that the **named** server software chose for itself is longer than the maximum allowed length.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: This is an unrecoverable error. You must contact the IBM Software Support Center. This error message is sufficient documentation.

Source Data Set: NDMAIN

Procedure Name: main

EZZ9572I *reason*

Explanation: This message can be issued for a number of different reasons. All of them pertain to the processing of a received NOTIFY request.

reason can be any of the following:

1. **notify question section empty** indicates that a malformed NOTIFY request was received that did not contain a question section.
2. **notify question section contains multiple RRs** indicates that a malformed NOTIFY request was received that contained more than one question in the question section.
3. **notify question section contains no SOA** indicates that a malformed NOTIFY request was received that did not contain the required SOA record in the question section.
4. **received notify for zone** *zone_name***: not authoritative** indicates that a NOTIFY request was received for zone, *zone_name*, but this **named** server is not authoritative for that zone.

System Action: The **named** server continues. The NOTIFY request is discarded.

User or Operator Response: Notify the system programmer.

System Programmer Response: Reasons 1, 2, or 3 indicate that a software problem in the software that is sending the NOTIFY request, or the sending software product, is incompatible with BIND **named** servers or this version of BIND. Reason 4 indicates a configuration error in either this **named** server or the **named** server sending the NOTIFY. The sending **named** server is configured assuming that this **named** server operates as a secondary **named** server for *zone_name*, but this **named** server is not configured to support that configuration. Contact the system programmer responsible for the **named** server sending the NOTIFY and resolve the configuration error.

Source Data Set: NOTIFY

Procedure Name: notify_log

EZZ9573I set maximum *option* **to** *value***:** *error_text*

Explanation: This message is issued when an attempt to specify a maximum resource limit for *option* is not accepted by the operating system.

option is the named.conf resource-limit option that is not accepted by the operating system.

value is the unaccepted value.

error_text contains more information about the error. When *error_text* is **invalid file**, the value is not allowed, either because it exceeds system limits, or falls below a threshold that is already been reached.

System Action: The **named** server continues. The resource limit for *option* for this process is not changed from the system defined value.

User or Operator Response: If *value* exceeds system limits, contact the System Programmer about raising the corresponding values in BPXPRMxx. If the value falls below a threshold that has already been reached, you may want to try stopping and restarting the **named** server. If this process does not have the authority to set system limits, contact the System Programmer. See the *z/OS MVS System Commands* for the description of the **DISPLAY OMVS,LIMITS** command and for more information about displaying current resource usage and system resource limits.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: set_limit

EZZ9574I port value *port_number* **is out of range**

Explanation: The port number specified on a **listen-on** statement does not fit within the allowed range.

port_number is the port number that is not within the allowable range.

System Action: The **named** server continues. The **listen-on** statement with the incorrect port number is ignored. The **named** server will not provide service to any adapters covered by this **listen-on** statement.

User or Operator Response: Contact the system programmer.

System Programmer Response: Specify a valid port number on the **listen-on** statement that is in error in the named.conf file and reload the **named** server.

Source Data Set: SERVER

Procedure Name: ns_listenelt_fromconfig

EZZ9575I 'max-cache-size *value***' is too large**

Explanation: The maximum cache size specified on the **max-cache-size** option in named.conf is larger than the maximum value allowed.

value is the maximum cache size value that is specified on the **max-cache-size** option in named.conf

System Action: The **named** server ends.

User or Operator Response: Change the **maximim-cache-size** value in named.conf so it is within the allowed range, and restart the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_view

EZZ9576I port *port_number* **out of range**

Explanation: A port number specified on a forwarders statement was not within the allowed range of values.

port_number is the value that is out of the valid range.

System Action: The **named** server ends.

User or Operator Response: Change the port value in error so that it is within the allowed range of port numbers and restart the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_forward

EZZ9577I no forwarders seen; disabling forwarding

Explanation: No IP addresses were found on a **forwarders** statement.

System Action: The **named** server continues. Forwarding is disabled.

User or Operator Response: Add one or more IP addresses to the **forwarders** statement in error, or remove the **forwarders** statement if there was no intention to forward DNS requests. Reload the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_forward

EZZ9578I zone '*zone_name***' 'type' not specified**

Explanation: The type of zone was not specified on the **zone** statement in error.

zone_name is the name of the zone appearing on the **zone** statement in error.

System Action: The **named** server ends.

User or Operator Response: Add the zone type to the zone statement in error and restart the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_zone

EZZ9579I option 'directory' contains relative path '*directory_path***'**

Explanation: The **directory** option contains a relative path instead of an absolute path.

directory_path is the relative path found by the program.

System Action: The **named** server continues.

User or Operator Response: A relative path name is valid; however, it might cause problems if the named.conf file is moved in the future. You should change the directory path to an absolute path.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: directory_callback

EZZ9580I no source of entropy found

Explanation: A **random-device** statement was not found in named.conf.

System Action: The **named** server continues.

User or Operator Response: This is an informational message. Lack of a **random-device** statement might cause dynamic updates to DNSSEC zones to fail, or might cause the automatic generation of the name.key file to fail.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: load_configuration

EZZ9581I could not open entropy source *random_device*: *error_text*

Explanation: The device specified on the **random-device** statement could not be opened.

random_device is the device or file that could not be opened.

error_text gives more information about the error.

System Action: The **named** server continues.

User or Operator Response: Make sure the file or device specified on the **random-device** statement in named.conf exists. If it exists, make sure the file or device can be opened for read access. Reload the **named** server when the problem has been corrected.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: load_configuration

EZZ9582I loading new zones failed: *error_text*

Explanation: An error was detected while attempting to add new zones to the **named** server with the **rndc reconfig** command.

error_text gives more information about the error.

System Action: The **named** server continues. The new zones added to the named.conf file will not be loaded by the **named** server.

User or Operator Response: Correct the error in the new zones added to named.conf and reissue the **rndc reconfig** command.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: reconfig

EZZ9583I isc_mutex_init() failed: *error_text*

Explanation: An attempt to acquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error.

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DNSCACHE

Procedure Name: dns_cache_create

EZZ9584I key '*key_id*'**: the only supported algorithm is hmac-md5:** *file_name*:*line_number*

Explanation: A **key** statement with a key using an unsupported algorithm for this function was found.

key_id is the name of the key containing the error.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: The key algorithm specified on the **algorithm** phrase of the **key** statement must be

changed to **hmac-md5**. Remove the key from the named.conf file, regenerate an hmac-md5 key and define it in the named.conf file. See the *z/OS Communications Server: IP System Administrator's Commands* for information about the dnssec-keygen command. Then restart the **named** server.

System Programmer Response: None.

Source Data Set: TSIGCONF

Procedure Name: add_initial_keys

EZZ9585I *source_file*: **file does not end with newline**

Explanation: An unexpected end-of-file condition caused the **named** server to end the parsing of the master file.

source_file is the name of the master file being processed.

System Action: The **named** server continues.

User or Operator Response: Ensure that the master file data being processed is not a partial file.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: WARNUNEXPECTEDEOF

EZZ9586I **dns_master_load:** *file_name*:*line_number*: **isc_lex_gettoken() failed:** *error_text*

Explanation: The **named** server failed to load a master file because the master file resource record token being processed was not one of the following:

- end-of-line
- end-of-file
- multiple line condition
- escape character

The master file being processed will not be loaded.

file_name is the name of the master file where the error was encountered.

line_number is the line number in the master file where the error was encountered.

error_text describes the error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the master file being loaded contains valid resource records.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: gettoken

EZZ9587I '*identity*' **is not a valid name:** *file_name*:*line_number*

Explanation: The **identity** field of an **update-policy** statement in named.conf is not a valid DNS name or wildcard name.

identity is the **identity** field in error of the **update-policy** statement.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: Specify a valid name for the **identity** field of the **update-policy** statement in error and restart the **named** server.

System Programmer Response: None.

Source Data Set: ZONECONF

Procedure Name: configure_zone_ssutable

EZZ9588I '*name*' **is not a valid name:** *file_name*:*line_number*

Explanation: The **name** field of an **update-policy** statement in named.conf is not a valid DNS name or wildcard name.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: Specify a valid name for the **name** field of the **update-policy** statement in error and restart the **named** server.

System Programmer Response: None.

Source Data Set: ZONECONF

Procedure Name: configure_zone_ssutable

EZZ9589I '*type*' **is not a valid type:** *file_name*:*line_number*

Explanation: The **type** field of an **update-policy** statement is not one of the valid types of resource records.

type is the **type** field in error of the **update-policy** statement.

file_name is the name of the configuration file where the error was encountered.

line_number is the line number in the configuration file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: The **type** field must be a valid DNS resource record type. Specify a valid type for the field and restart the **named** server. Refer to the description of the **update-policy** statement in the *z/OS Communications Server: IP System Administrator's Commands* for information about valid DNS resource record types.

System Programmer Response: None.

Source Data Set: ZONECONF

Procedure Name: configure_zone_ssutable

EZZ9590I **dns_master_load:** *source_file*:*line_number*: **ignoring out-of-zone data (***name***)**

Explanation: The **named** server found records that belong to another zone. This record is ignored and the master file processing continues.

source_file is the name of the master file being processed.

line_number is the line number in the master file.

name is the domain name of the resource record in error.

System Action: The **named** server continues.

User or Operator Response: Ensure that the record in error is associated with the correct zone. After correcting the error, reload the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9591I *source_file*:*line_number*: **no TTL specified; using SOA MINTTL instead**

Explanation: The **named** server accepts (with a warning) master files beginning with an SOA record without an explicit TTL field and lacking a $TTL directive, by using the SOA MINTTL as a default TTL. This is for backwards compatibility with old versions of BIND 8, which accepted such files without warning although they are illegal according to RFC 1035. See Appendix B, “Related protocol specifications (RFCs)” on page 733 for information about accessing RFCs.

source_file is the master file where the error was encountered.

line_number is the line number in the master file where the error was encountered.

System Action: Processing continues.

User or Operator Response: Define an explicit TTL field on the affected SOA record or add a $TTL directive to the zone master file.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9592I *source_file*:*line_number*: **no TTL specified; zone rejected**

Explanation: Zone loading is rejected because a resource record, which is not an SOA, has no defined TTL, and there is no $TTL directive to specify a default TTL for the zone records.

source_file is the master zone file where the error was encountered.

line_number is the line number in the master zone file where the error was encountered.

System Action: Processing continues without loading the affected zone.

User or Operator Response: Update zone file with $TTL directive or specify TTL value on every resource record where such value is not specified.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9593I retrieving the address of an IPv6 socket from the kernel failed.

Explanation: The **named** server attempted an AF_INET6 getsockname() C Runtime call, but the call failed, indicating that IPv6 is not enabled on the TCP/IP stack on which the **named** server is running.

System Action: The **named** server continues. All communication over IPv6 in the **named** server is disabled.

User or Operator Response: Contact the system programmer.

System Programmer Response: The **named** server is unable to successfully issue the getsockname() C Runtime call. This call is issued to test whether IPv6 support is enabled on the system. Check to make sure that an IPv6 enabled TCP/IP stack is running on this system. After an IPv6 enabled TCP/IP stack has been started, restart the **named** server.

Source Data Set: NET

Procedure Name: try_proto

EZZ9594I IPv6 support is disabled.

Explanation: The **named** server will no longer use IPv6 for any communications.

System Action: The **named** server continues. IPv4 communications will still be enabled unless specifically disabled by the listen-on option in named.conf.

User or Operator Response: Notify the system programmer.

System Programmer Response: This message is issued with EZZ9593I indicating that IPv6 is not enabled on this

system. Ensure that an IPv6 enabled TCP/IP stack is running on this system. After an IPv6 enabled TCP/IP stack has been started, the **named** server will automatically detect and use its IPv6 interfaces.

Source Data Set: NET

Procedure Name: try_proto

EZZ9595I could not set master file style

Explanation: The **named** server failed to convert the resource data set or question section because of a problem with the master file style. The data being processed might be too long or there might be insufficient buffer space to convert the data from column format.

System Action: The **named** server continues.

User or Operator Response: Ensure the master file data being process is correct. If you are unable to resolve this problem, then contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: MASTERDU

Procedure Name: dns_master_dumpnodetostream

EZZ9596I internal_accept(): accept() returned peer address family *peer_family* **(expected** *expected_peer_family***)**

Explanation: The **named** server failed to accept a client connection because of an incorrect socket address family.

peer_family is the family of the peer address.

expected_peer_family is the expected socket peer family.

System Action: The **named** server continues.

User or Operator Response: Contact the systems programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the **TC**, **UDP** and **IOCTL** options and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: internal_accept

EZZ9599I setsockopt(*socket_descriptor***, IPV6_V6ONLY) failed:** *errno***, errno2=***errnojr*

Explanation: A setsockopt call with the IPv6_V6ONLY option failed.

socket_descripitor is the socket descriptor of the socket affected by the failing call.

errno is the hexadecimal UNIX System Services Return Code. These return codes are listed and described in the *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 Code section of the *z/OS UNIX System Services Messages and Codes*, where the Reason Codes are listed.

System Action: The **named** server continues. IPv4 data sent to the **named** server might be received on an IPv6 socket as an IPv4-mapped IPv6 address. Consequently, any named.conf options that refer to the pure IPv4 address form (in an address match list, for example) will not take effect on the intended data because the **named** server is using an IPv4-mapped IPv6 address instead of the pure IPv4 address.

User or Operator Response: Notify the system programmer.

System Programmer Response: Ensure that the TCP/IP stack is IPv6 enabled in the BPXPRMxx SYS1.PARMLIB member.

Source Data Set: ISCSOCKT

Procedure Name: isc_socket_create

EZZ9600I SNTP server ready

Explanation: The Simple Network Time Protocol (SNTP) server started successfully.

System Action: The SNTP server is waiting for requests from SNTP clients.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sntpd.c

Procedure Name: main()

EZZ9601I SNTP server ended

Explanation: The Simple Network Time Protocol (SNTP) server ended. This might be caused by a kill command from the UNIX shell, a stop command from the MVS console, or an internal error.

System Action: SNTP ends

User or Operator Response: Restart the SNTP server.

System Programmer Response: None.

Source Data Set: sntpd.c

Procedure Name: main()

EZZ9602I SNTP server initializing

Explanation: The Simple Network Time Protocol (SNTP) server is initializing.

System Action: SNTP server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: sntpd.c

Procedure Name: main()

EZZ9650I VIPADISTRIBUTE WITH SYSPLEXPORTS REJECTED FOR DVIPA *ip_addr*

Explanation: A SYSPLEXPORTS keyword was found on a VIPADISTRIBUTE statement for a Dynamic VIPA (DVIPA) that already had a VIPADISTRIBUTE statement specified without the SYSPLEXPORTS keyword. The SYSPLEXPORTS keyword must be specified on the first VIPADISTRIBUTE statement specified for a specific DVIPA.

ip_addr is the IP address specified on the VIPADISTRIBUTE statement containing the rejected SYSPLEXPORTS keyword.

System Action: Processing continues. The VIPADISTRIBUTE statement is rejected.

User or Operator Response: Contact your system programmer.

System Programmer Response: To enable SYSPLEXPORTS you must first delete all previous VIPADISTRIBUTE statements for this DVIPA. Then reissue the VIPADISTRIBUTE with SYSPLEXPORTS specified.

Source Data Set: EZBXFDYN

Procedure Name: ValidateVDIST

EZZ9651I TCPSTACKSOURCEVIPA IP ADDRESS *ip_addr* **WAS NOT USED**

Explanation: An outbound connection request was processed but the IP address configured with TCPSTACKSOURCEVIPA could not be used as the source address because the address was not in the stack Home List, and did not fall within a configured VIPARANGE so a Dynamic VIPA could not be created.

ip_addr is the IP address specified on the TCPSTACKSOURCEVIPA parameter on the IPCONFIG statement. To avoid flooding the system console, this informational message will not be issued again for at least five minutes.

System Action: TCP/IP continues. The local address of the socket will be the address of the physical interface, unless there is a static VIPA in the HOME list above the physical link. If so, the static VIPA will be used.

User or Operator Response: Contact your systems programmer

System Programmer Response: Change the TCPSTACKSOURCEVIPA address to an IP address in the stack's HOME list, or an IP address that falls within a configured VIPADYNAMIC VIPARANGE.

Source Data Set: EZBXFUT2

Procedure Name: EZBXFSSV

EZZ9680I IPv6 structures in kernel and user space do not match

Explanation: The compile-time and run-time sizes of the sockaddr_in6 C program structure differ.

System Action: The **named** server continues. All IPv6 support in the **named** server is disabled.

User or Operator Response: Contact the system programmer.

System Programmer Response: Verify that you are running the IBM Communications Server for z/OS on the same version of z/OS with which it was shipped. If you believe this is the case, contact the IBM Software Support Center with this message description.

Source Data Set: NET

Procedure Name: try_proto

EZZ9681I internal_accept(): accept() failed to return remote address

Explanation: An accept() C Runtime call completed successfully but the remote (peer) address that is normally returned by the call was not present.

System Action: The **named** server continues. The **named** server is unable to verify that the connected socket is with the expected client.

User or Operator Response: Contact the system programmer.

System Programmer Response: Re-create this problem with a SYSTCPIP component trace active specifying the TC option and set the debug level on the **named** log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs
- The component trace requested

Source Data Set: ISCSOCKT

Procedure Name: internal_accept

EZZ9682I dns_master_load: *file_name*:*line_number*: **$INCLUDE not allowed**

Explanation: A $INCLUDE directive was found but is not allowed in this file.

file_name is the name of the file where the $INCLUDE was found.

line_number is the line number in the file where the $INCLUDE was found.

System Action: The **named** server will end.

User or Operator Response: Place the data from the $INCLUDE file directly into the file where the $INCLUDE directive appeared and remove the $INCLUDE directive. Restart the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9683I dns_master_load: *error_text*

Explanation: The **named** server failed to successfully parse and load a zone from a zone data file because of insufficient memory.

error_text gives more information about the error.

System Action: The **named** server ends.

User or Operator Response: Notify the system programmer.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: MASTER

Procedure Name: load

EZZ9684I dns_master_load: *file_name*:*line_number*: *dns_name*: *error_text*

Explanation: The **named** server failed to successfully parse and load a zone because of a problem with a resource record or set of resource records.

file_name is the name of the master zone file being processed at the time of the error.

line_number is the line number of the master zone file where the error was encountered.

dns_name is the owner name of the resource record in error.

error_text gives more information about the error.

If *error_text* is **CNAME and other data**, then the master zone file contains a CNAME and at least one other resource record with the same owner name as the CNAME resource record, which is not allowed.

System Action: The **named** server ends.

User or Operator Response: Correct the error in the master zone file and restart the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: commit

EZZ9685I no_references: dns_rbt_deletenode: *error_text*

Explanation: The call to the procedure, **dns_rbt_deletenote** to delete a node from the procedure, **no_references** failed because of an internal error. The failing call was to remove a node from the **named** server's internal Red-Black Tree Database.

error_text gives more information about the error.

System Action: The **named** server continues.

User or Operator Response: This is an internal error. Provide a storage dump of the **named** server address space and contact the IBM Software Support Center.

System Programmer Response: None.

Source Data Set: RBTDB

Procedure Name: no_references

EZZ9686I lame server resolving '*dns_name***' (in '***domain_name***'?):** *ip_address*

Explanation: The **named** server received a response from a **named** server at *ip_address* while attempting to resolve the name, *dns_name* in the domain, *domain_name*. The parent **named** server of the *domain_name* **named** server incorrectly delegated the *domain_name* domain to the **named** server at *ip_address*, and the **named** server at *ip_address* is not authoritative for the domain, *domain_name*. That server is termed a 'lame server' because of the misconfiguration.

dns_name is the name that was being resolved by the **named** server.

domain_name is the domain name that was incorrectly delegated.

ip_address is the address of the **named** server that is not actually authoritative for *domain_name*, but is believed to be authoritative by the **named** server that is authoritative for *domain_name*'s parent domain.

System Action: The **named** server continues. The name, *dns_name* will not be resolved.

User or Operator Response: Notify the DNS administrator of the **named** server authoritative for the parent domain of *domain_name* that it incorrectly delegated the *domain_name* domain to the wrong **named** server.

System Programmer Response: None.

Source Data Set: RESOLVER

Procedure Name: log_lame

EZZ9687I transfer of '*zone_name***/***class***' from** *ip_address***:** *message_text*

Explanation: This message is issued during an incoming zone transfer for multiple situations, which are described by *message_text*. This message might be issued as an error, warning, or informational message.

zone_name is the zone that is being transferred to this **named** server.

class is the class of the zone that is usually **IN**.

ip_address is the TCP/IP address this **named** server is receiving the zone transfer from.

message_text describes why this message was issued

System Action: In all cases, the **named** server continues. If the severity of the message is **error**, the zone transfer failed. If the severity of the message is **warning** the zone transfer might have partially succeeded. If the severity is **info** the zone transfer might or might not have succeeded.

User or Operator Response: If the zone transfer failed, contact the DNS administrator of the master server for *zone_name* to correct the problem indicated by *message_text*.

System Programmer Response: None.

Source Data Set: XFRIN

Procedure Name: xfrin_logv

EZZ9688I zone *zone_name***:** *message_text*

Explanation: The **named** server uses this message to log zone processing errors. If the severity of the message is not **informational**, this message should be preceded by another message with more information about the error.

zone_name is the name of the zone being processed.

message_text describes the error or informational message.

System Action: The **named** server continues.

User or Operator Response: If the message severity is not informational, correct the error described in the *message_text* and any previous error messages. Reload or stop and restart the **named** server.

System Programmer Response: None.

Source Data Set: ZONE

Procedure Name: dns_zone_log

EZZ9689I zone *zone_name*: *message_text*

Explanation: The **named** server uses this message to log information related to the NOTIFY process between master and slave **named** servers of a particular zone.

zone_name is the name of the zone being processed.

message_text describes the informational message.

System Action: The **named** server continues. The NOTIFY process proceeds normally.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ZONE

Procedure Name: notify_log

EZZ9690I isc_condition_init() failed

Explanation: The **named** server failed to initialize a conditional variable (using pthread_cond_init()) because of insufficient memory.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Increase the region size and restart the **named** server.

Source Data Set: TASK

Procedure Name: isc_taskmgr_create

EZZ9691I no matching 'forwarders' statement

Explanation: The **forward** option was specified without specifying a **forwarders** list.

System Action: The **named** server ends.

User or Operator Response: Notify the system programmer.

System Programmer Response: Either remove the **forward** option from the named.conf file, or add a **forwarders** option with a list of IP addresses, then restart the **named** server.

Source Data Set: CHECK

Procedure Name: check_forward

EZZ9692I *option* '*option_value*' **is out of range**

Explanation: The value coded on the specified option statement is greater than the maximum allowed value.

option is the named.conf option in error.

option_value is the value coded on the option.

System Action: The **named** server ends.

User or Operator Response: Code a valid value for the option indicated in the message and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference* for information about the option and its valid values.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_options

EZZ9693I zone '*zone_name*': type not present

Explanation: No zone type was specified on the **type** clause of the **zone** statement.

zone_name is the **zone** statement without a zone type.

System Action: The **named** server ends.

User or Operator Response: Specify a valid zone type after the **type** clause of *zone_name* in named.conf and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9694I zone '*zone_name*': invalid type *zone_type*

Explanation: The zone type specified on the **type** clause of the **zone** statement is not valid.

zone_name is the **zone** statement with a zone type that is not valid.

zone_type is the unrecognized zone type coded on the **type** clause of the **zone** statement.

System Action: The **named** server ends.

User or Operator Response: Specify a valid zone type on the **type** clause of the **zone** statement in error and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference* for the valid zone types.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9695I zone '*zone_name*': already exists

Explanation: The zone was defined more than once in named.conf.

zone_name is the zone that appears more than once in named.conf.

System Action: Only the first occurrence of *zone_name* will be loaded. The **named** server continues.

User or Operator Response: One of the **zone** definitions with the name *zone_name* should be renamed to a unique zone name within named.conf or deleted if it is a duplicate. Restart or reload the **named** server after making the change.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9696I option '*option_name*' is not allowed in '*zone_type*' zone '*zone_name*'

Explanation: The specified option is incompatible with *zone_type*.

option_name is the incompatible option that was specified.

zone_type is the type of zone on which the incompatible option was specified.

zone_name is the zone name where the incompatible option was found.

System Action: The **named** server ends.

User or Operator Response: Notify the system programmer.

System Programmer Response: Remove the *option_name* from *zone_name* and restart the **named** server.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9697I zone '*zone_name*': missing 'masters' entry

Explanation: A zone statement was encountered that was either of type **slave** or **stub**, and no **masters** clause was present within the zone statement.

zone_name is the zone statement in error.

System Action: The **named** server ends.

User or Operator Response: Specify a masters clause on the zone statement in error and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9698I zone '*zone_name*': 'allow-update' is ignored when 'update-policy' is present

Explanation: The **allow-update** and **update-policy** statements in the zone, *zone_name* cannot coexist.

zone_name is the zone statement in error.

System Action: The **named** server ends.

User or Operator Response: Remove the **update-policy** statement or the **allow-update** statement from the affected zone and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9699I dialup type '*dialup_type*' is not allowed in '*zone_type*' zone '*zone_name*'

Explanation: The **dialup** option specified is incompatible with the zone type in which it is specified.

dialup_type is the type of dialup option coded in the zone statement.

zone_type is the type of zone in which the incompatible dialup option was found.

zone_name is the name of the zone where the error occurred.

System Action: The **named** server ends.

User or Operator Response: Change the dialup type, remove the **dialup** statement, or change the zone type of the affected zone and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9700I invalid dialup type '*dialup_type*' in zone '*zone_name*'

Explanation: The dialup type specified on the **dialup** option is not a valid value.

dialup_type is the dialup type that is not valid.

zone_name is the zone where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: Change the dialup type to a valid value and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_zoneconf

EZZ9701I key '*key_name*': already exists

Explanation: The key was already defined.

key_name is the name of the key in error.

System Action: The **named** server ends.

User or Operator Response: Remove or rename the duplicate key and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: check_viewconf

EZZ9702I key '*key_name*' must have both 'secret' and 'algorithm' defined

Explanation: A **key** statement is missing either the **algorithm** clause or the **secret** clause, or both.

key_name is the **key** statement in error.

System Action: The **named** server ends.

User or Operator Response: Add the missing clause to the **key** definition and restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: cfg_check_key

EZZ9703I when using 'view' statements, all zones must be in views

Explanation: A **zone** statement was found outside a **view** statement. If any **view** statements are present in the named.conf file, all **zone** statements must appear within a **view** statement.

System Action: The **named** server ends.

User or Operator Response: Move the **zone** statement inside an appropriate **view** statement, or remove all of the **view** statements within named.conf, and then restart the **named** server.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: cfg_check_namedconf

EZZ9704I dns_master_load: *source_file*:*line_number*: **SOA record not at top of zone (***zone_name***)**

Explanation: The zone name of the SOA record did not match the zone name of the **zone** statement in the named.conf file.

source_file is the file where the error was detected.

line_number is the line number in the file where the error was detected.

zone_name is the **zone** statement in the named.conf file.

System Action: The **named** server continues. The zone data in *source_file* might be ignored.

User or Operator Response: Either change *zone_name* on the **zone** statement in named.conf or change the **zone** name of the SOA record in *source_file* so that they match. Reload the **named** server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9705I 'cache-file' cannot be a global option if views are present

Explanation: The **cache-file** option may not be present under the **options** statement when **view** statements are present in the named.conf file.

System Action: The **named** server ends.

User or Operator Response: Remove the **cache-file** option from the **options** statement, and add the **cache-file** statement to the appropriate **view** statements.

System Programmer Response: None.

Source Data Set: CHECK

Procedure Name: cfg_check_namedconf

EZZ9706I dns_master_load: *file_name*:*line_number*: *include_file_name*: *error_text*

Explanation: An error occurred processing the $INCLUDE directive.

file_name is the name of the file that includes *include_file_name*.

line_number is the line number in *file_name* where *include_file_name* is included.

include_file_name is the name of the file on the $INCLUDE directive.

error_text gives more information about the error.

System Action: The **named** server continues. Data in the **included** file will not become a part of the zone data.

User or Operator Response: Use the *error_text* information to correct any errors in the name server zone data file or the **included** file. Ensure that the file to be included exists, is not corrupted, and the permission bits allow the file to be read. If there are too many open files, try to reduce the number of nested $INCLUDE statements or the number of zone files.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9707I isc_socket_listen() failed: *error_text*

Explanation: A socket listen() call failed unexpectedly while attempting to set up a socket for the rndc control channel. Message EZZ9399I should precede this error message and it will give more specific information about the failure.

error_text gives more information about the error.

System Action: The **named** server continues. This **named** server will not be able to accept rndc commands on the affected socket.

User or Operator Response: Notify the system programmer.

System Programmer Response: This error is either caused by an internal **named** server error, or a shortage of memory. Use message EZZ9399I to determine the reason. If this error was not caused by a lack of memory, re-create this problem with a SYSTCPIP component trace active specifying the TC, UDP, and IOCTL options and set the debug level on the **named** server log files to 99. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** server logs
- The component trace requested

Source Data Set: CNTLCONF

Procedure Name: control_listen

EZZ9708I isc_socket_recv() failed: *error_text*

Explanation: A recvmsg() call failed while processing an inbound UDP DNS request.

error_text gives more information about the error

System Action: The **named** server continues. The affected DNS request will be reprocessed a short time later. The time to reprocess the request should be unnoticeable to the end user.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: CLIENT

Procedure Name: client_udprecv

EZZ9709I Unexpected label type *label_type*

Explanation: The program failed to create a file name from a DNS name because the DNS name contained a label that was longer than 63 characters and was not a bitstring label.

label_type is the non-bitstring label type contained in the DNS name

System Action: The program does not complete successfully.

User or Operator Response: In most cases, the label is a domain name. If so, shorten the domain name and try the program again.

System Programmer Response: None.

Source Data Set: DNSNAME

Procedure Name: dns_name_tofilenametext

EZZ9710I could not construct absolute path of configuration file: *error_text*

Explanation: When the **named** server was started, a relative path was given for the named.conf file. The **named** server attempted to construct the absolute path by prefixing the current directory, but was unsuccessful.

error_text gives more information about the error.

System Action: The **named** server ends.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the constructed absolute path name does not exceed 1023 characters. If the constructed absolute path name exceeds this limit, move the named.conf file to a directory closer to the root directory, or shorten the directory names.

Source Data Set: NDMAIN

Procedure Name: setup

EZZ9711I could not set up forwarding for domain '*domain_name***':** *error_text*

Explanation: An internal error occurred while processing the statements that allow forwarding for the domain, *domain_name*. The most likely cause is a storage shortage.

domain_name is the DNS domain name for which forwarding was being set up when the error occurred.

error_text gives more information about the error.

System Action: The **named** server ends.

User or Operator Response: Notify the system programmer.

System Programmer Response: If *error_text* indicates a shortage of memory, increase region size and restart the **named** server. If the problem persists or if *error_text* does not indicate a memory problem, re-create this problem with the debug level on the **named** log files to three. Obtain the following documentation and contact the IBM Software Support Center:

- The **named** server configuration file
- The SYSLOG DAEMON logging file
- All configured **named** logs

Source Data Set: SERVER

Procedure Name: configure_forward

EZZ9712I 'additional-from-cache no' is only supported with 'recursion no'

Explanation: A value of **no** for the **additional-from-cache** option is incompatible when recursion is disabled.

System Action: The **named** server continues.

User or Operator Response: The **additional-from-cache** option should be enabled in the named.conf file, or recursion should be enabled with the **recursion** option in named.conf. After the changes have been made, reload the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_view

EZZ9713I could not configure root hints from '*hints_file***': *error_text***

Explanation: The file containing the root hints could not be loaded.

hints_file is the root hints file that was being processed at the time of the error.

error_text contains more information about the error.

System Action: The **named** server ends.

User or Operator Response: Ensure that *root_hints* exists in the working directory and file permissions allow the file to be read. Verify that the file has the proper format and syntax. If *error_text* indicates a storage shortage, notify the system programmer about the error.

System Programmer Response: If *error_text* indicates a storage shortage, increase the region size and restart the **named** server.

Source Data Set: SERVER

Procedure Name: configure_zone

EZZ9714I dns_master_load: *source_file*:*line_number*: signature has expired

Explanation: A SIG (signature) resource record expired according to the system clock. When signatures are created, they have an expiration time associated with them. When that time has elapsed, the signature is considered to be expired.

source_file is the file containing the signature that expired.

line_number is the line number in the file where the expired signature was found.

System Action: The **named** server continues. The resource record data associated with the expired signature might not be usable.

User or Operator Response: The zone file containing the expired signature should be re-signed using the dnssec-signzone tool. After re-signing the zone file, reload the name server and allow sufficient time for slave servers to synchronize the data with the master server.

System Programmer Response: None.

Source Data Set: MASTER

Procedure Name: load

EZZ9715I *source_file*:*source_line*: **warning** "*ip_address*" **is not a decimal dotted quad**

Explanation: The IP address is not a valid IPv4 address.

source_file is the file where the error was found.

source_line is the line number in the file where the error was found.

ip_address is the IP address that is not valid.

System Action: The **named** server continues. The resource record containing the error will not be usable.

User or Operator Response: Use *source_file* and *source_line* information to find and change the IP address to be a valid IPv4 address. Reload the **named** server.

System Programmer Response: None.

Source Data Set: RDATA

Procedure Name: getquad

EZZ9716I **the key** '*key_name*' **is too short to be secure**

Explanation: The **named** server configuration file defines a key that is fewer than 64 bits in length. Keys smaller than 64 bits are insecure and should not be used.

System Action: The **named** server continues. The key remains usable.

User or Operator Response: In order to provide adequate security, ensure that the key used is at least 64 bits in length.

System Programmer Response: None.

Source Data Set: TSIG

Procedure Name: dns_tsigkey_createfromkey

EZZ9717I **dumping master file:** *tempfile_name*: **fsync:** *description*

Explanation: The **named** server could not write to the temporary dump output file.

tempfile_name is the the unique temporary file being closed.

description describes the error.

System Action: The **named** server deletes the temporary dump file and continues.

User or Operator Response: Ensure that there is adequate space on the output device to contain the entire zone database being dumped.

System Programmer Response: None.

Source Data Set: MASTERDU

Procedure Name: dns_master_dump

EZZ9718I **extra data in root hints** '*file_name*'

Explanation: The root hints file might not contain the necessary information to locate the root zone **named** servers, or it might contain extraneous data. The root hints file should contain only NS resource records for the root zone and A resource records corresponding to those root **named** servers.

file_name is the name of the root hints file where the error was discovered.

System Action: The **named** server continues. The extra data will be ignored.

User or Operator Response: Examine the root hints file, *file_name* and remove all resource records except the NS and A records needed to find the root **named** servers. Reload the **named** server.

System Programmer Response: None.

Source Data Set: ROOTNS

Procedure Name: dns_root_create

EZZ9719I *file_name*:*line_number*:*error_text*

Explanation: A syntax error was encountered.

file_name is the file where the error was found.

line_number is the line number in the file where the error was found.

error_text gives specific information about the error.

System Action: Messages with a severity level of WARNING will allow the **named** server to continue. Messages with a severity level of ERROR will cause the **named** server to end. The severity of the message can be displayed in the **named** server logging file with the message if the **print-severity** option is enabled in the **channel** phrase of the **logging** statement in named.conf.

User or Operator Response: Use the *error_text* information to correct the error in *file_name* at or near the line, *line_number*. Restart or reload the **named** server.

System Programmer Response: None.

Source Data Set: PARSER

Procedure Name: parser_complain

EZZ9720I **'max-buffered-messages'** *option_value* **is too large:** *file_name*:*line_number*

Explanation: The value specified for the **max-buffered-messages** option is larger than the largest allowable value.

option_value is the value that was specified on the **max-buffered-messages** option in named.conf. This value is not allowed.

file_name is the file where the error was encountered.

line_number is the line number in the file where the error was encountered.

System Action: The **named** server ends.

User or Operator Response: See the *z/OS Communications Server: IP Configuration Reference* for the allowable values for max-buffered-messages. Choose an allowable value and restart the **named** server.

System Programmer Response: None.

Source Data Set: SERVER

Procedure Name: configure_view

EZZ9721I **unknown class** '*class*'

Explanation: An unknown class was discovered on a **zone** or **view** statement.

class is the unknown **class** discovered while reading the **named** server configuration file.

System Action: The **named** server ends.

User or Operator Response: Specify a valid class for the view or zone, and restart the **named** server. See the **zone** statement in the *z/OS Communications Server: IP Configuration Reference* for valid classes.

System Programmer Response: None.

Source Data Set: CONFIG

Procedure Name: ns_config_getclass

EZZ9722I **omitting IPv4 interface** *interface_name* **from localnets ACL:** *error_text*

Explanation: The network mask for the specified interface was incorrect. The most likely cause is that the network mask is not contiguous.

interface_name is the interface with the incorrect network mask.

error_text gives more information about the error.

System Action: The **named** server continues. The interface in error will not be included in the built-in Access Control List (ACL), localnets.

User or Operator Response: Notify the system programmer.

System Programmer Response: If the TCP/IP stack configuration specified a network mask for this interface, correct the network mask so that the resulting network is a valid IPv4 network. Apply the changes to the TCP/IP stack and either restart the **named** server or wait for the **named** server **interface-interval** to expire so the TCP/IP interfaces will be re-scanned by the **named** server. Then check the **named** server logs for errors.

Source Data Set: INTRFCEM

Procedure Name: do_ipv4

EZZ9723I logging channel '*channel_name*' **file** '*file_name*'**:** *error_text*

Explanation: The file, *file_name* specified in the *channel_name* logging channel, could not be opened.

channel_name is the channel where the error was encountered.

file_name is the name of the file that could not be opened.

error_text gives more information about the error.

System Action: The **named** server continues. Logging to the channel, *channel_name* will not occur.

User or Operator Response: Verify that the path for *file_name* exists, that the directory and file permission bits will allow the **named** server to write to the file, and that there is space available on the file system. Restart the **named** server after correcting the errors.

System Programmer Response: None.

Source Data Set: LOGCONF

Procedure Name: channel_fromconf

EZZ9724 isc_mutex_init() failed: *error_text*

Explanation: An attempt to acquire a mutually exclusive lock failed because of insufficient memory or inadequate security to create a lock.

error_text describes the error

System Action: If the **named** server is initially loading, it ends. If the **named** server is reloading, it stops reloading and continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Ensure that the **named** server user ID has sufficient authority to perform locking. Increase the region size and restart the **named** server. See the *z/OS Communications Server: IP Configuration Reference*.

Source Data Set: DNSCACHE

Procedure Name: cache_cleaner_init

EZZ9780I DUPLICATE ADDRESS DETECTED FOR *type* **CONFIGURED HOME ADDRESS** *addr* **ON INTERFACE** *interface*

Explanation: During the verification of the home address for this node, another node on the link indicated that it was already using the address. The home address cannot be used.

type specifies how the home address was configured, and is one of the following:

MANUAL lindicates that the home address was manually configured in the TCP/IP profile.

AUTO Indicates that the address is either a link-local automatically generated address or an automatically configured address.

addr specifies the address that is a duplicate.

interface is the interface name.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Another node is already using the address. TCP/IP cannot use this address if the other node is using the same address. If this is a manually configured address, you can update the TCP/IP profile to specify a different address and issue a VARY TCPIP,,OBEYFILE command to activate the new profile. If this is an automatically configured address, duplicate interface identifiers might be assigned to multiple interfaces. Ensure that each interface has a unique interface identifier.

Source Data Set: EZB6PDAD, EZB6PNBR

Procedure Name: DAX_otherLifs, EZB6PAHA, EZB6PANA

EZZ9781I AUTOCONFIGURED ADDRESS *addr* **EXPIRED ON INTERFACE** *interface*

Explanation: The valid lifetime for the autoconfigured address expired and can no longer be used.

addr specifies the address that is no longer valid.

interface is the interface name.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: No action is required if the address was expected to expire. Otherwise, configure the router to indicate the valid lifetime for this address.

Source Data Set: EZB6PNBR

Procedure Name: EZB6PRHA

EZZ9782I ALTERNATE LINK LOCAL ADDRESS *addr* **ON INTERFACE** *interface* **FAILED**

Explanation: A full, manually configured address was configured for the specified interface. The link-local address generated by using the hardware-provided interface ID failed the duplicate-address detection process. A randomly generated interface ID was tried and also failed the duplicate-address detection process. If this is the first randomly generated interface ID, a second randomly generated interface ID will be tried. No more than two randomly generated interface IDs will be tried.

addr is the alternate address generated using a random interface ID.

interface is the interface name.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Message EZZ9780I indicates the original link-local address that failed duplicate-address detection. Check the OSA setup to determine why its interface ID is not unique. If both attempts with alternate interface IDs failed, restarting the interface again might generate a unique ID successfully. If subsequent attempts fail with the same error, consider bypassing duplicate-address detection (by specifying DUPADDRDET 0 on the INTERFACE statement) until the source of the problem can be resolved.

Source Data Set: EZB6PDAD, EZB6PNBR

Procedure Name: Fail_DAD, AHA_tryrandomifcid

EZZ9783I ACTIVATION OF ADDRESS *addr* **ON** *interface* **STOPPED -** *reason*

Explanation: During the activation of a device, an error occurred that prevented the completion of the activation process.

addr is the address that was not activated.

interface is the interface name.

reason indicates the source of the problem and is one of the following:

STORAGE NOT AVAILABLE Indicates that storage could not be obtained to create the required control blocks.

SOLICITED NODE JOIN FAILED Indicates that the multicast solicited node join failed.

ALL NODES JOIN FAILED Indicates that the multicast all nodes join failed.

LINK LOCAL ALREADY EXISTS Indicates that a link-local address had been requested but an active or activating link-local address already exists.

System Action: The address was not activated. TCP/IP continues.

User or Operator Response: None.

System Programmer Response: If STORAGE COULD NOT BE OBTAINED is the reason, resolve storage constraints before restarting the interface. For other reasons, stopping and restarting the interface might resolve the problem.

Source Data Set: EZB6PDAD, EZB6PNBR

Procedure Name: EZB6PDAX, EZB6PANA

EZZ9784I ALTERNATE LINK LOCAL ADDRESS *addr* **ON INTERFACE** *interface* **SUCCESSFUL**

Explanation: A full, manually configured address was configured for the specified interface. The link-local address generated by using the hardware-provided interface ID failed the duplicate-address detection process. A randomly generated interface ID was tried and successfully completed the duplicate-address detection process.

addr is the alternate address generated by using a random interface ID.

interface is the interface name.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: Message EZZ9780I indicates that the original link-local address failed duplicate-address detection. Check the OSA setup to determine why its interface ID is not unique.

Source Data Set: EZB6PDAD

Procedure Name: EZB6PDAX

EZZ9785I PURGECACHE FAILED FOR *name* **-** *reason*

Explanation: The PURGECACHE request could not be processed for one of the reasons listed below.

name is the name specified on the vary command.

name is one of the following:

LINK OR INTERFACE DOES NOT EXIST
The link or interface name does not exist.

CACHE NOT SUPPORTED
The link or interface does not support an ARP or neighbor cache.

OSA FLUSH SUPPORT NOT AVAILABLE
The installed level of OSA does not support the ARP Assist Flush subcommand. The cache was not purged. This error applies to OSA outboard ARP cache entries only. The minimum IPv4 OSA-Express microcode level required for this support is GA3.

STORAGE NOT AVAILABLE
CSM storage required to complete the PURGECACHE request could not be obtained.

System Action: TCPIP continues.

User or Operator Response: If the error reason is **LINK OR INTERFACE DOES NOT EXIST** or **CACHE NOT SUPPORTED**, verify that the correct link or interface name was specified.

System Programmer Response: If the error reason is **OSA FLUSH SUPPORT NOT AVAILABLE**, a new level of the OSA microcode might be needed if purging of outboard ARP cache is required.

If the error reason is **STORAGE NOT AVAILABLE**, issue the D NET,CSM command and determine if sufficient CSM storage has been defined.

Source Data Set: EZB6PNBC

Procedure Name: EZB6PPRG

EZZ9786I PURGECACHE PROCESSED FOR *type name*

Explanation: The PURGECACHE request was processed. If the cache is maintained by OSA (for example, IPv4 QDIO), the request to purge the ARP cache has been sent to OSA to perform. Otherwise, the local cache for the requested link or interface has been purged.

type indicates whether the name matches a link or interface. *type* can be either LINK or INTERFACE.

name is the link name or interface name of the cache that was purged.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZB6PNBC

Procedure Name: EZB6PPRG

EZZ9787I ERROR *error_code* **REGISTERING MULTICAST ADDRESS** *mcast_address* **FOR DEVICE** *device_name*

Explanation: The adapter reported an error while attempting to register a multicast address with the device.

error_code is the error code reported by the adapter.

mcast_address is the multicast address.

device_name is the name of the device.

System Action: TCPIP continues, but is unable to receive any packets destined for this multicast address over this device.

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

System Programmer Response: See the *zSeries: OSA-Express Customer's Guide and Reference* for information about OSA Reject Codes and a description of the error code.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ9788I ERROR *error_code* **REGISTERING MULTICAST ADDRESS** *mcast_address* **FOR INTERFACE** *interface_name*

Explanation: The adapter reported an error while attempting to register a multicast address with the interface.

error_code is the error code reported by the adapter.

mcast_address is the multicast address.

interface_name is the name of the interface.

System Action: TCPIP continues, but is unable to receive any packets destined for this multicast address over this interface.

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

System Programmer Response: See the *zSeries: OSA-Express Customer's Guide and Reference* for information about OSA Reject Codes and a description of the error code.

Source Data Set: TCPIP

Procedure Name: EZBIFIND

EZZ9789I ALTERNATE INTERFACE ID *newid* **ASSIGNED TO INTERFACE** *interface_name*

Explanation: The Interface ID generated from the OSA-provided data is mapped to a reserved value and is replaced with an alternate value. The alternate interface ID will not be preserved across an interface restart. If an interface is stopped and restarted, autoconfigured addresses on the interface will change.

newid is the alternate Interface ID.

interface_name is the name of the interface.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZB6PNBR

Procedure Name: EZB6PAHA

EZZ9790I MBDATACONN incorrect syntax. Specify MBDATACONN=(*file_system_cp***,***network_transfer_cp***). MBDATACONN ignored.**

Explanation: A LOCSITE subcommand was entered with an MBDATACONN parameter that has incorrect syntax. The MBDATACONN parameter must be a pair of codepage names.

file_system_cp is the name of the file system codepage.

network_transfer_cp is the name of the network transfer codepage.

System Action: FTP continues.

User or Operator Response: Reissue the LOCSITE subcommand with a valid value for the MBDATACONN parameter. See the *z/OS Communications Server: IP User's Guide and Commands* for information about the parameters of the LOCSITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

EZZ9791I MBDATACONN parameter is too long. Maximum length for code page name is *length***. MBDATACONN ignored.**

Explanation: A LOCSITE subcommand was entered with an MBDATACONN parameter specifying a code page name that is too long. The MBDATACONN parameter is ignored.

length is the maximum name length allowed.

System Action: FTP continues.

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

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

EZZ9792I No conversion available to *cp_name1* from *cp_name2*. MBDATACONN ignored.

Explanation: A LOCSITE subcommand was entered with the MBDATACONN parameter, but there is no supported code set converter for the code sets that are specified. The MBDATACONN parameter is ignored.

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: FTP continues.

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

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

EZZ9793I Multi-byte encoding requested but codepages are not defined.

Explanation: A file transfer subcommand was issued and the client program determined that a multi-byte data transfer was needed because the data type was ASCII and ENCODING=MBCS was coded in FTP.DATA or was specified on a LOCSITE subcommand. However, no MBDATACONN value was specified.

System Action: The requested transfer subcommand failed.

User or Operator Response: Specify the MBDATACONN parameter on a LOCSITE subcommand or specify the MBDATACONN statement in the FTP.DATA file. See the *z/OS C/C++ Programming Guide* for information about supported code set converters and valid code set names. Resubmit the data transfer subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCG EZAFTPCP

Procedure Name: checkMB

EZZ9794I Multi-byte encoding requires a structure of FILE

Explanation: A file transfer subcommand was issued and the current structure is not FILE.

System Action: The requested transfer subcommand failed.

User or Operator Response: Specify the FILE subcommand and resubmit the data transfer subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCG EZAFTPCP

Procedure Name: checkMB

EZZ9795I Multi-byte encoding requires a mode of STREAM

Explanation: A file transfer subcommand was issued and the current transfer mode is not STREAM.

System Action: The requested transfer subcommand failed.

User or Operator Response: Specify the STREAM subcommand and resubmit the data transfer subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCG EZAFTPCP

Procedure Name: checkMB

EZZ9796I Multi-byte encoding requires FILETYPE=SEQ

Explanation: A file transfer subcommand was issued and the current filetype is not SEQ.

System Action: The requested transfer subcommand failed.

User or Operator Response: Specify the FILETYPE=SEQ parameter on a LOCSITE subcommand or specify the FILETYPE SEQ statement in the FTP.DATA file

System Programmer Response: None.

Source Data Set: EZAFTPCP

Procedure Name: checkMB

EZZ9797I Multi-byte encoding not supported for RECFM=*recfm*

Explanation: A file transfer subcommand was issued and the file that is sent from or received into is an MVS data set.

recfm is the record format (RECFM) of the data set. The only record formats that are allowed for multi-byte encoded data transfers are V, VB, and U.

System Action: The requested transfer subcommand fails

User or Operator Response: If the subcommand is a get, and the local data set is new, use the LOCSITE subcommand with either RECFM=V, RECFM=VB, or RECFM=U to cause FTP to create a data set with the correct format. Multi-byte encoding is not supported if the existing data set is not record format V, VB, or U.

System Programmer Response: None.

Source Data Set: EZAFTPCG EZAFTPCP

Procedure Name: mvs_rcvFile setFileAccessStructF

EZZ9798I Multi-byte encoding not supported for the SRESTART subcommand

Explanation: An SRESTART subcommand was entered and multi-byte encoding processing is also requested. A stream mode restart requires that FTP count the bytes that were successfully transferred. FTP cannot correctly count bytes transferred when multi-byte conversions are occurring.

System Action: The SRESTART subcommand failed.

User or Operator Response: Restart the data transfer without the SRESTART subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCB

Procedure Name: srestart

EZZ9799I Multi-byte encoding not supported when transferring RDWs

Explanation: A data transfer subcommand was entered for a variable format data set while the option to send record descriptor words (RDWs) was also in effect. The FTP client cannot send RDWs and also do multi-byte encoding.

System Action: The requested subcommand failed.

User or Operator Response: Issue a LOCSITE NORDW subcommand and resubmit the data transfer subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCP

Procedure Name: setFileAccessStructF

EZZ9809I Multi-byte encoding does not support *codepage* as a file system codepage. MBDATACONN ignored.

Explanation: The format of the MBDATACONN parameter is the following:

```
MBDATACONN=(file_system_cp,network_transfer_cp)
```

The multi-byte encoding support for the FTP client 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: FTP continues.

User or Operator Response: Resubmit the subcommand with one of the supported file system codepages.

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

EZZ9810I Multi-byte encoding does not support *codepage* as a network transfer codepage. MBDATACONN ignored.

Explanation: The format of the MBDATACONN parameter is the following:

```
MBDATACONN=(file_system_cp,network_transfer_cp)
```

The multi-byte encoding support for the FTP client 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: FTP continues.

User or Operator Response: Resubmit the subcommand with the supported network transfer codepage.

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

EZZ9811I ENcoding parameter *parm* is not valid. Encoding ignored.

Explanation: The LOCSITE subcommand was issued with the ENcoding parameter, but the value *parm* specified for the parameter is not valid. The value specified for ENcoding should be SBCS or MBCS.

System Action: The ENcoding parameter is ignored. FTP continues.

User or Operator Response: Reissue the LOCSITE subcommand with a valid value for the ENcoding parameter. See the *z/OS Communications Server: IP User's Guide and Commands* for information about the parameters of the LOCSITE subcommand.

System Programmer Response: None.

Source Data Set: EZAFTPCK

Procedure Name: locsite

Chapter 10. SNM*xxxxx* messages

SNM000W MSG *number* does not exist in SNMPMSGS message table DSISNMnn

Explanation: SNMP attempted to display the indicated message but could not find this message number in the DSIMSG data set.

System Action: The missing message is not displayed. Processing continues.

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

System Programmer Response: Check that the data set containing SNMP messages (members DSISNMxx) is properly specified on the DSIMSG DD statement of the NetView start procedure.

Source Data Set: SNMPMSGS

Procedure Name: snmpmsgs

SNM010E *command* Invalid netmask or desired network specified

Explanation: SNMP encountered an error converting an IP address specified in a command. This message indicates that the netmask or network specified is incorrect.

System Action: The command is ignored.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct network and netmask specified. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct network and netmask.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM011E *command* Invalid function: *function*

Explanation: An SNMP command specified an incorrect function.

System Action: The command is ignored.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct function. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct function.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM012E *command* Function *function* needs at least *count* parameter(s)

Explanation: An SNMP command omitted the required function parameters.

System Action: The command is abnormally ended.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct parameters. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct parameters.

Source Data Set: SNMPSNMP

Procedure Name: snmp

© Copyright IBM Corp. 1994, 2002

SNM013E *command* **Missing function**

Explanation: An SNMP command omitted the required function specification.

System Action: The command is ignored.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct function specified. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct function.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM014E *command* **Missing variable name**

Explanation: An SNMP command omitted a variable name.

System Action: The command is abnormally ended.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct variable name specified. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct variable name.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM015E *command* **Missing set value for variable**

Explanation: SNMP found a SET function with no value specified.

System Action: The command is abnormally ended.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct value for the set variable. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct value for the set variable.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM016E *command* **Missing or invalid destination host identification**

Explanation: An SNMP command omitted a required host ID specification.

System Action: The command is abnormally ended.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct host ID specified. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct host ID.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM017E *command* **Missing community name**

Explanation: An SNMP command omitted a required community name specification.

System Action: The command is ignored.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct community name specified. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct community name.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM018E *command* **Request rejected,** *task* **task not active**

Explanation: An SNMP request was rejected because the referenced SNMP IUCV task was not active.

System Action: SNMP requests are rejected until the SNMPIUCV task is activated.

User or Operator Response: Start the SNMPIUCV task first, wait for the connection with the SNMP Query Engine, and reissue the SNMP request.

System Programmer Response: None.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM019E *command* **Request rejected,** *task* **task not (yet) connected**

Explanation: An SNMP request was rejected because the indicated task had not yet established an IUCV connection with the SNMP Query Engine.

System Action: The connection is still pending. The system continues operation.

User or Operator Response: Wait for the connection to complete, and reissue the SNMP request.

System Programmer Response: None.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM020E *command* **Error** *error* **on DSIGET**

Explanation: SNMP encountered an error allocating memory for a packet.

System Action: The command ends abnormally. The system tries to continue operation.

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

System Programmer Response: Use the indicated error value to determine the reason for the error. The cause of this error is probably insufficient storage. See *NetView Customization: Using Assembler* for DSIGET error codes.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM021E *command* **Error** *error* **on DSIMQS**

Explanation: SNMP encountered an error queuing a packet to the SNMP IUCV task.

System Action: The command ends abnormally. The system tries to continue operation.

User or Operator Response: Retry the command. If the error persists, tell the system programmer about the error.

System Programmer Response: Use the indicated error value to determine the reason for the error. See *NetView Customization: Using Assembler* for DSIMQS error codes.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM022E *command* **Missing or invalid trap filter id**

Explanation: An SNMP command failed to specify or specified an incorrect trap filter ID.

System Action: The command ends abnormally.

User or Operator Response: If the command was issued at the NetView command line, reenter the command with the correct trap filter ID specified. If the command was issued from a NetView CLIST, ask the system programmer to correct the CLIST.

System Programmer Response: If the command was issued from a NetView CLIST, correct the NetView CLIST to specify the correct trap filter ID.

Source Data Set: SNMPSNMP

Procedure Name: snmp

SNM023W **Error** *code* **on DSIDKS** *service*, **DDname is** *ddname* **- using internal defaults**

Explanation: The indicated NetView disk service received the indicated return code while accessing a data set specified on the indicated DD statement in the NetView start procedure. This error occurs when SNMPIUCV cannot access the SNMPARMS member of the DSIPARM data set.

System Action: The SNMPARMS member is not read. Internal default values are used for the parameters contained in SNMPARMS.

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

System Programmer Response: Check that the data set containing the SNMPARMS member is properly specified on the DSIPARM DD statement in the NetView start procedure. See *NetView V1R3 Customization: Using Assembler* for information about DSIDKS services and return codes.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM030I **SNMP request** *filter_id* **received the following trap:**

Explanation: The indicated SNMP trap filter received a trap. This message is the first of a series of messages (SNM030I–SNM039I) that contain the detailed trap information. The indicated filter ID is associated with the SNMP TRAPSON request that set the trap filter.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM031I **Agent Address:** *ip_address*

Explanation: The indicated ip address is the address of the SNMP agent that issued the trap. This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM032I Generic trap type: *type*

Explanation: The indicated trap type is the generic trap type of the received trap. Refer to *z/OS Communications Server: IP User's Guide and Commands* for a list of generic trap types and their meanings. This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM033I Specific trap type: *type*

Explanation: The indicated trap type is the specific trap type of the received trap. This value is 0 unless the generic trap type was 6 (enterprise-specific trap). In the case of an enterpriseSpecific trap, see the documentation for the host system of the agent issuing the trap for information about enterprise-specific trap types. This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM034I Time stamp: *time*

Explanation: The indicated time is the time at which the SNMP agent generated the trap. The time is expressed in hundredths of a second and gives the elapsed time since the SNMP agent was started (the sysUptime value). This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM035I Enterprise Object ID: *object_id*

Explanation: The indicated object ID is the object ID of the agent that issued the trap. This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM036I Variable name: *name*

Explanation: The indicated name is the asn.1 name of a variable that was included with the trap to provide additional information for the trap. This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM037I Variable value type: *type*

Explanation: The indicated type is the syntax for the variable in SNM036I. The type is one of the following values:

Value	Meaning
0	Text representation
1	Number (integer, signed)
2	Binary data string
3	Object identifier
4	Empty (no value)
5	Internet address
6	Counter (unsigned)
7	Gauge (unsigned)
8	Timeticks (1/100ths of a second)
9	Display string

This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM038I Variable value: *value*

Explanation: The indicated value gives the value of the variable in SNM036I. This message is part of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM039I SNMP Request *filter_id* **End of trap data**

Explanation: This message indicates the end of the trap message series beginning with SNM030I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM040I **Request** *request* **from** *operator* **Returned the following response:**

Explanation: A response was received for the indicated request, which was issued by the indicated NetView operator ID. The request number matches the request number displayed by message SNM050I when the request was issued. Message SNM040I is the first of a series of messages (SNM040I–SNM049I) that contain information from the response.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM042I **Variable name:** *name*

Explanation: The indicated name is the asn.1 name of the variable for which information is returned. This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM043I **Variable value type:** *type*

Explanation: The indicated type is the syntax for the variable for which information is being returned. The type is one of the following values:

Value	Meaning
0	Text representation
1	Number (integer, signed)
2	Binary data string
3	Object identifier
4	Empty (no value)
5	Internet address
6	Counter (unsigned)
7	Gauge (unsigned)
8	Timeticks (1/100ths of a second)
9	Display string

This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM044I Variable value: *value*

Explanation: The indicated value gives the value of the variable for which information is being returned. This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM045I Major Error code: *code*

Explanation: The indicated code is the major error code contained in the response. The major error code is one of the following:

Code	Meaning
0	No error was detected
1	The SNMP agent reported the error
2	The error was detected internally

This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: See SNM048I for information about the various major and minor error codes.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM046I Minor Error code: *code*

Explanation: The indicated code is the minor error code contained in the response. The meaning of the minor error code is dependent on the value of the major error code. The minor error code can have the following values:

Agent detected (major error code 1)

Code	Meaning
0	No error was detected
1	Too big
2	No such name
3	Bad value
4	Read only
5	General error

Internally detected (major error code 2)

Code	Meaning
0	No error was detected
1	Protocol error
2	Out of memory
3	No response – all retries failed
4	Some I/O error occurred
5	Illegal request
6	Unknown host specified
7	Unknown MIB variable
8	No such filter
9	Too many variables specified

This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: See SNM048I for information about the various major and minor error codes.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM047I Error Index: *index*

Explanation: If the SNMP agent detected an error, SNM047I indicates the position of the first variable in error. For example, if a GET request is issued with a list of four variables, and the major or minor error code indicates an agent detected error, the error index shows which of the four variables (1, 2, 3, or 4) is in error. This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: See SNM048I for information about the various major and minor error codes.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM048I Error Text: *error*

Explanation: This message gives a textual description of the error code in message SNM045I and SNM046I. The following is a list of errors and actions:

Bad value:
A Set request was issued for a variable, but the value specified to set the variable was out of range or had incorrect syntax.

General error:
The agent detected an error other than one of the indicated errors. If this message was received in response to a Get request for one of the enterprise-specific 3172 variables at a VM or MVS SNMP agent, the TCP/IP virtual machine console (on VM) or address space output file (on MVS) should have additional error messages.

I/O error:
An Input/Output error occurred. If this message appeared in response to a PING request, check that the name of the SNMP query engine address space is authorized to use RAW sockets.

Illegal request:
An incorrect request was received. Contact the IBM Software Support Center.

No error:
The request was completed successfully with no detected errors.

No response:
The SNMP query engine sent a request to an SNMP agent, but no response was received. Check that the route to the specified host is available and that the SNMP agent was started.

No such filter:
An SNMP TRAPSOFF command was issued for a trap filter ID, but the specified trap filter ID does not exist.

No such name:
The SNMP request sent to the agent contained a request for a variable that the SNMP agent does not support.

Out of memory:
The SNMP query engine address space ran out of memory. Restart SQESERV with a larger region size. If the error persists, contact the IBM Software Support Center.

Protocol error:
A protocol error occurred. Contact the IBM Software Support Center.

Read only:
A Set request was issued for a variable, but the SNMP agent did not allow write access to the variable.

Too big:
When the SNMP agent was generating the SNMP response PDU, the total length of all the requested variables and their values was too large for the response PDU. This happens when multiple variables, each with a large value, are requested on the same SNMP request PDU. To recover from this error, break the SNMP request into many smaller requests, and reissue the SNMP commands.

Too many elements:
An SNMP command was issued with more than 10 MIB variables requested. A maximum of 10 variable names and values can be specified in one SNMP command. Issue the variable names using multiple SNMP commands instead of one.

Unknown host:
The SNMP command specified a host name that could not be resolved into an IP address. Check that the host name was entered correctly. Check that the SNMP query engine can reach the domain name server or has access to the HOSTS.INFO data set (whichever is used for name resolution). Check that the host name is specified correctly at the domain name server or in the HOSTS.LOCAL data set.

Unknown variable:
An SNMP command was entered with a MIB variable that was specified in textual form. The variable could not be found in the *hlq*.MIB@DESC.DATA data set. Check that the variable name was entered correctly. Add the variable to the *hlq*.MIB@DESC.DATA data set, if necessary.

This message is part of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: See the previous description.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM049I SNMP Request *request* **End of response**

Explanation: This message indicates the end of the response message series beginning with SNM040I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM050I SNMP Request *request* **from** *operator* **accepted, sent to Query Engine**

Explanation: The SNMP command that the indicated operator just entered was accepted by the SNMP command processor and passed to the SNMP query engine. This request was assigned the indicated request number. The same request number is displayed in SNM040I when the response is received and is used to correlate incoming responses with outstanding requests.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM051E SNMP Request *request* **from** *origin* **discarded, MsgComplete IPAUDIT=X***value*

Explanation: SNMP received an IUCV Message Complete notification, indicating the message was discarded rather than received. Execution continues. The indicated origin identifies the operator that issued the request, and the indicated request identifies the request that this operator issued.

System Action: The system tries to continue operation but discards the request that failed.

User or Operator Response: Check that the SNMP Query Engine is operating correctly, and reissue your SNMP request. If the error persists, tell the system programmer about the error.

System Programmer Response: Determine why the IUCV Message was not received by the Query Engine. The IPAUDIT code can be used to determine the cause. See *VM CP Programming Services* for more information about the IPAUDIT value. Correct the situation, and reissue the SNMP request.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM052E SNMP Request *request* **from** *originator* **discarded, IUCV SEND error (R15)=***error*

Explanation: SNMP encountered the indicated error while attempting an IUCV SEND. Consequently, the indicated request from the indicated user could not be processed and was discarded.

System Action: The system continues to operate.

User or Operator Response: Check that the SNMP Query Engine is operating correctly. Check that the SNMPIUCV task is connected to the SNMP Query Engine. Reissue your SNMP request. If the error persists, tell the system programmer about the error.

System Programmer Response: Determine the cause of the error by analyzing the error code. See *VM CP Programming Services* for more information about the error code on the IUCV SEND. Correct the situation, and reissue the SNMP request.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM053E SNMP Request *request* **from** *originator* **discarded, IUCV Connection Severed**

Explanation: The indicated SNMP request from the indicated operator was discarded because the SNMP IUCV connection was severed.

System Action: Because the connection was severed, no new SNMP requests are accepted until the connection with the SNMP Query Engine is reestablished.

User or Operator Response: Reestablish the IUCV connection with the SNMP Query Engine and reissue your SNMP request.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM100I SNMP task *task* **connecting to Query Engine** *name* **path=***path*

Explanation: The SNMP task running under NetView is trying to connect to the SNMP query engine. The indicated name is the name of the query engine address space. The indicated path is the IUCV path ID that the connection is using.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM101W SNMP task *task* **found Query Engine** *name* **not ready**

Explanation: SNMP attempted to connect to the Query Engine but found the Query Engine was not ready. The indicated name identifies the started task of the Query Engine.

System Action: Depending on the reason that the engine is not ready, SNMP can end this attempt abnormally to contact the engine or continue execution. The SNMP task *task* retries the connection to the Query Engine *name* based on the retry value specified in the SNMP initialization parameters.

User or Operator Response: Start the SNMP query engine. If the Query Engine cannot be started, or if the error persists, tell the system programmer about the error.

System Programmer Response: Check the SNMP Query Engine log for error messages. Check that the value of the SNMPQE parameter in the SNMPARMS parameter list corresponds to the name of the SNMP Query Engine address space.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM102I SNMP task *task* **connected to Query Engine** *name*, **path=***path*, **msglim=***msglim*

Explanation: The indicated SNMP task running under NetView successfully connected to the SNMP query engine. The indicated name is the name of the SNMP query engine address space. The indicated path ID is the IUCV path ID being used. The indicated msglim is the IUCV MSGLIM.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM103E SNMP task *task* **got error** *error* **on** *function sub-function*

Explanation: SNMP encountered the indicated error from the indicated IUCV function.

System Action: The request is abnormally ended. If possible, the system continues operation.

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

System Programmer Response: Use the indicated error code, function, and subfunction values to determine the cause of the error. See *VM CP Programming Services* for more information about IUCV error codes. Correct the situation, and reissue the command.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM104E SNMP task *task* **has** *count* **IUCV interrupts pending, path severed**

Explanation: SNMP received more IUCV interrupts than it can handle with its current preallocated queue. The indicated number of IUCV interrupts are pending. The IUCV path was severed.

System Action: The system tries to reestablish the connection with the SNMP Query Engine and restart.

User or Operator Response: Tell the system programmer about the error. You can reissue your SNMP requests after the connection with the SNMP Query Engine has been reestablished.

System Programmer Response: Check that you do not have a CLIST that issues SNMP requests in a endless loop. If the error persists, contact the IBM Software Support Center.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM105E SNMP task *task* **ignored IUCV interrupt, unexpected IPTYPE** *type*

Explanation: SNMP received an IUCV interrupt with an unexpected IPTYPE field.

System Action: The interrupt is ignored. The system continues operation.

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

System Programmer Response: Check that the version and level of SNMP Query Engine are in sync with the version of the SNMPIUCV task.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM106E SNMP task *task* **connection to Query Engine** *id* **now severed**

Explanation: The IUCV connection between SNMP and the indicated query engine was severed. The connection can be reopened.

System Action: If any requests are still outstanding, the system discards them. Execution continues.

User or Operator Response: Reestablish the connection with the SNMP Query Engine if SNMP requests need to be issued.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM107E SNMP task *task* **received sever for path=***path*

Explanation: SNMP received notification of a severed IUCV connection for the indicated path. SNMP did not have this connection open before this notification.

System Action: The system continues operation.

User or Operator Response: If the error persists, contact the system programmer.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM108E SNMP task *task* **cannot find IUCV PC numbers in CVT.**

Explanation: SNMP could not locate the IUCV Program Call numbers.

System Action: Initialization is terminated. The SNMPIUCV task cannot operate in this situation.

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

System Programmer Response: Check that TCP/IP was installed correctly on your MVS system.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM109I SNMP task *task* **awaiting Connection Complete from Query Engine** *name*

Explanation: The SNMP task running under NetView sent a request for a connection to the SNMP query engine, but the query engine has not yet responded. The indicated name is the name of the query engine address space.

System Action: The indicated SNMP task waits for the connection complete message from the query engine.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM120W SNMP task *task* **got error** *error* **on DSIGET**

Explanation: SNMPIUCV encountered an error allocating memory with a DSIGET call.

System Action: During initialization, this error can prevent the startup of the SNMPIUCV task. During normal operation, this error can cause one or more SNMP requests to fail, or never receive a response.

User or Operator Response: If the error persists, tell the system programmer about the error.

System Programmer Response: Use the indicated error value to determine the reason for the error. The probable cause of the error is not enough storage. See *NetView Customization: Using Assembler* for DSIGET error codes. If the error indicates not enough storage, start NetView with a larger region size.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM121W SNMP task *task* **incoming packet has invalid packet length** *length*

Explanation: SNMP received a packet with an incorrect length.

System Action: Incorrect packets are ignored.

User or Operator Response: If the error persists, tell the system programmer about the error.

System Programmer Response: Check that you have a correct level of the SNMP Query Engine that is in sync with the SNMPIUCV task on NetView side.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM122W SNMP task *task* **incoming packet has invalid character set** *value*

Explanation: SNMP received a packet specifying an incorrect character set.

System Action: The packet is ignored. The system continues operation.

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

System Programmer Response: Check that the version and level of SNMP Query Engine are in sync with the version of the SNMPIUCV task.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM123W SNMP task *task* **incoming packet has invalid packet type** *value*

Explanation: SNMP received a packet with an incorrect packet type.

System Action: The packet is ignored. The system continues operation.

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

System Programmer Response: Check that the version and level of SNMP Query Engine are in sync with the version of the SNMPIUCV task.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM124I SNMP task *subtask* **received: Connection** *condition*

Explanation: The state of the IUCV connection between the indicated NetView subtask and the SNMP query engine changed to either Quiesced or Resumed. If the NetView subtask is sending more requests than the query engine can handle, the query engine halts the connection. The requests are queued at the NetView subtask. When the query engine is ready for more requests, it resumes the connection, and the NetView subtask sends the requests again.

System Action: When the connection is quiesced, the NetView subtask accepts requests from the operator but queues the request rather than forward the requests to the query engine. When the connection resumes, the NetView subtask begins forwarding queued and new requests to the query engine again.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM128W SNMP task *task* **found missing value or invalid value for** *parameter*

Explanation: SNMP found an initialization parameter with a missing or incorrect value.

System Action: The missing or incorrect value is ignored, and the default value is used. Initialization continues.

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

System Programmer Response: Correct the missing or incorrect value in SNMPARMS.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

SNM129W SNMP task *task* **found invalid initialization parameter** *value*

Explanation: SNMP found an incorrect initialization parameter.

System Action: The incorrect initialization parameter is ignored. Initialization continues and assumes the default value for the incorrect parameter.

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

System Programmer Response: Correct the incorrect initialization parameter in SNMPARMS.

Source Data Set: SNMPTASK

Procedure Name: snmpiucv

Appendix A. Message ranges and the components that issue them

Volume 1 ranges

Number Span	Component
EZA0519 - EZA0661	TCP/IP Utilities
EZA1450 - EZA2947	FTP
EZA2950 - EZA3172	Network Computing
EZA3950 - EZA4173	Network Database
EZA4250 - EZA4257	Portmapper
EZA4275 - EZA4361	Remote Procedure Call
EZA4375 - EZA4427	REXEC/D
EZA4726 - EZA4809	REXEC
EZA4986 - EZA5049	RSH
EZA5125 - EZA5651	SMTP
EZA5775 - EZA6230	SNALINK & SNALU6.2
EZA6275 - EZA6635	SQESERVE
EZA8200 - EZA8417	Telnet
EZA8500 - EZA9744	Common SayCalRe
EZAIN11	

Volume 2 ranges

Number Span	Component
EZB0600 - EZB0857	LPD
EZB0900 - EZB1100	LPT
EZB1200 - EZB1230	X Window
EZB2000 - EZB2498	XNX25
EZB2500 - EZB2785	Retired — do not use
EZB3000 - EZB3566	DNS
EZB3825 - EZB4198	NCPROUTE
EZB6473	Initialization
EZB8801	FFST

Volume 3 ranges

Number Span	Component
EZY0002 - EZY0058	Packet Trace
EZY0620 - EZY0974	Network Print facility
EZY1020 - EZY1035	IMS Socket
EZY1120 - EZY1130	Sockets Extended

© Copyright IBM Corp. 1994, 2002

Number Span	Component
EZY1218 - EZY1348	CICS
EZY1372 - EZY1424	REXX Socket
EZY1870 - EZY2145	MVS Platform
EZY2370 - EZY2443	MISCSRV
EZY2632 - EZY2832	FTP
EZY3720 - EZY4199	Sockets
EZY4200 - EZY4484	Telnet DBCS
EZY5398	MVS Platform
EZY6001 - EZY6077	Restartable VMCF
EZY6101 - EZY6261	FFS Writer for JES
EZYFS01 - EZYFS95 EZYFT01 - EZYFT99	FTP
EZYPR001 - EZYPR214	Network Print Facility ISPF
EZYRC01 - EZYRC24	OE REXEC
EZYRD01 - EZYRD39	OE REXECD
EZYRP35 - EZYRP72 (EZA4331 - EZA4335)	OE RPC
EZYRS01 - EZYRS66	OE RSHD
EZYTE01 - EZYTE87 EZYTO01 - EZYTO12 EZYTS02 - EZYTS18 EZYTU01 - EZYTU36 EZYTY01 - EZYTY14	OS/390 UNIX Telnet
EZYXM01 - EZYXR42	OE OSF/Motif API
EZYXU01 - EZYXU99	OE OSF/Motif User Interface Language Compiler
EZYXW01 - EZYXW69	OE x Window

Volume 4 ranges

Number Span	Component
EZZ0053 - EZZ0720	Configuration & Initialization
EZZ2351 - EZZ2389	ONETSTAT & NETSTAT
EZZ3105 - EZZ3133	OPING
EZZ3200 - EZZ3249	SNMP Subagent
EZZ3250 - EZZ3255	Operator Command
EZZ3300 - EZZ3343	OSNMP
EZZ4200 - EZZ4221	Initialization
EZZ4300 - EZZ4348	Interface Layer
EZZ4350	Initialization
EZZ4360 - EZZ4367	Storage Utilization
EZZ4820 - EZZ5027	OE ROUTED
EZZ6002 - EZZ6084	Telnet
EZZ6201 - EZZ6316	SNMP Agent
EZZ6351 - EZZ6388	OTRACERT
EZZ6452 - EZZ6705	DNS

Number Span	Component
EZZ7001 - EZZ7049	TFTP
EZZ7051 - EZZ7062	TIMED
EZZ7251 - EZZ7331	DHCP
EZZ7350 - EZZ7448	NSM
EZZ7450 - EZZ7452	Initialization
EZZ7475 - EZZ7476	Stack Configuration
EZZ7500 - EZZ7738	Sendmail & Popper
EZZ7800 - EZZ8111	OMPROUTE
EZZ8201 - EZZ8229	SLA Subagent
EZZ8252 - EZZ8260	D TCPIP,,SYSPLEX
EZZ8301 - EZZ8314	VIPA
EZ8320 - EZZ8331	HOSTNAME, DOMAINNAME, DNSDOMAINNAME, HOST
EZZ8340 - EZZ8345	Misc. Command
EZZ8392 - EZZ8420	Trap Forwarder Daemon
EZZ8431 - EZZ8451	Service Policy Agent
EZZ8453 - EZZ8460	Storage Display
EZZ8461 - EZZ8477	Sysplex Distributer
EZZ8495- EZZ8503	Intrusion Detection Services (SYSLOG)
EZZ8504 - EZZ8511	Intrusion Detection Services Statistics (TRMDSTAT)
EZZ8540	Intrusion Detection Services (SYSLOG)
EZZ8600 - EZZ8631	DCAS
EZZ8638 - EZZ8653	Intrusion Detection Services SYSLOG
EZZ8761 - EZZ8769	Intrusion Detection Services Console
EZZ8771 - EZZ8774	Policy Agent (PAGENT) Messages
EZZ8791 - EZZ9290	DNS Bind 9
EZZ9291 - EZZ9304	System Resolver
EZZ9315	Initialization
EZZ9316 - EZZ9327	Intrusion Detection Services SYSLOG
EZZ9331 - EZZ9599	DNS Bind 9
EZZ9600 - EZZ9602	SNTP
EZZ9650 - EZZ9651	VIPA
EZZ9680 - EZZ9724	DNS Bind 9
EZZ9780 - EZZ9784	Neighbor Discovery
EZZ9790 - EZZ9811	FTP
SNM000 - SNM129	SNMPLIB

Appendix B. Related protocol specifications (RFCs)

This appendix lists the related protocol specifications for TCP/IP. The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. Some of these protocols are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement these particular functions or protocols. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

These documents can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

where:
nnnn Is the RFC number.
TXT Is the text format.
PS Is the PostScript format.

You can see Internet drafts at http://www.ietf.org/ID.html. See “Draft RFCs” on page 740 for draft RFCs implemented in z/OS V1R4 Communications Server.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC` *nnnn* for text versions or a subject line of `RFC` *nnnn*`.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil`.

Many RFCs are available online. Hard copies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available using FTP from the NIC at the following Web address: http://www.rfc-editor.org/rfc.html.

Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

Many features of TCP/IP Services are based on the following RFCs:

RFC	Title and Author
768	*User Datagram Protocol* J.B. Postel
791	*Internet Protocol* J.B. Postel
792	*Internet Control Message Protocol* J.B. Postel
793	*Transmission Control Protocol* J.B. Postel
821	*Simple Mail Transfer Protocol* J.B. Postel

© Copyright IBM Corp. 1994, 2002

822 *Standard for the Format of ARPA Internet Text Messages* D. Crocker

823 *DARPA Internet Gateway* R.M. Hinden, A. Sheltzer

826 *Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware* D.C. Plummer

854 *Telnet Protocol Specification* J.B. Postel, J.K. Reynolds

855 *Telnet Option Specification* J.B. Postel, J.K. Reynolds

856 *Telnet Binary Transmission* J.B. Postel, J.K. Reynolds

857 *Telnet Echo Option* J.B. Postel, J.K. Reynolds

858 *Telnet Suppress Go Ahead Option* J.B. Postel, J.K. Reynolds

859 *Telnet Status Option* J.B. Postel, J.K. Reynolds

860 *Telnet Timing Mark Option* J.B. Postel, J.K. Reynolds

861 *Telnet Extended Options—List Option* J.B. Postel, J.K. Reynolds

862 *Echo Protocol* J.B. Postel

863 *Discard Protocol* J.B. Postel

864 *Character Generator Protocol* J.B. Postel

877 *Standard for the Transmission of IP Datagrams over Public Data Networks* J.T. Korb

885 *Telnet End of Record Option* J.B. Postel

896 *Congestion Control in IP/TCP Internetworks* J. Nagle

903 *Reverse Address Resolution Protocol* R. Finlayson, T. Mann, J.C. Mogul, M. Theimer

904 *Exterior Gateway Protocol Formal Specification* D.L. Mills

919 *Broadcasting Internet Datagrams* J.C. Mogul

922 *Broadcasting Internet Datagrams in the Presence of Subnets* J.C. Mogul

950 *Internet Standard Subnetting Procedure* J.C. Mogul, J.B. Postel

952 *DoD Internet Host Table Specification* K. Harrenstien, M.K. Stahl, E.J. Feinler

959 *File Transfer Protocol* J.B. Postel, J.K. Reynolds

974 *Mail Routing and the Domain Name System* C. Partridge

1006 *ISO Transport Service on top of the TCP Version 3* M.T.Rose, D.E. Cass

1009 *Requirements for Internet Gateways* R.T. Braden, J.B. Postel

1011 *Official Internet Protocols* J. Reynolds, J. Postel

1013 *X Window System Protocol, Version 11: Alpha Update* R.W. Scheifler

1014 *XDR: External Data Representation Standard* Sun Microsystems Incorporated

1027 *Using ARP to Implement Transparent Subnet Gateways* S. Carl-Mitchell, J.S. Quarterman

1032 *Domain Administrators Guide* M.K. Stahl

1033 *Domain Administrators Operations Guide* M. Lottor

1034 *Domain Names—Concepts and Facilities* P.V. Mockapetris

1035 *Domain Names—Implementation and Specification* P.V. Mockapetris

1042 *Standard for the Transmission of IP Datagrams over IEEE 802 Networks* J.B. Postel, J.K. Reynolds

1044 *Internet Protocol on Network System's HYPERchannel: Protocol Specification* K. Hardwick, J. Lekashman

1055 *Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP* J.L. Romkey

1057 *RPC: Remote Procedure Call Protocol Version 2 Specification* Sun Microsystems Incorporated

1058 *Routing Information Protocol* C.L. Hedrick

1060 *Assigned Numbers* J. Reynolds, J. Postel

1073 *Telnet Window Size Option* D. Waitzman

1079 *Telnet Terminal Speed Option* C.L. Hedrick

1091 *Telnet Terminal-Type Option* J. VanBokkelen

1094 *NFS: Network File System Protocol Specification* Sun Microsystems Incorporated

1096 *Telnet X Display Location Option* G. Marcy

1101 *DNS encoding of network names and other types* P.V. Mockapetris

1112 *Host Extensions for IP Multicasting* S. Deering

1118 *Hitchhikers Guide to the Internet* E. Krol

1122 *Requirements for Internet Hosts—Communication Layers* R.T. Braden

1123 *Requirements for Internet Hosts—Application and Support* R.T. Braden

1155 *Structure and Identification of Management Information for TCP/IP-Based Internets* M.T. Rose, K. McCloghrie

1156 *Management Information Base for Network Management of TCP/IP-Based Internets* K. McCloghrie, M.T. Rose

1157 *Simple Network Management Protocol (SNMP)* J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin

1158 *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* M.T. Rose

1179 *Line Printer Daemon Protocol* The Wollongong Group, L. McLaughlin III

1180 *TCP/IP Tutorial* T.J. Socolofsky, C.J. Kale

1183 *New DNS RR Definitions* C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris, (Updates RFC 1034, RFC 1035)

1184 *Telnet Linemode Option* D. Borman

1187 *Bulk Table Retrieval with the SNMP* M.T. Rose, K. McCloghrie, J.R. Davin

1188 *Proposed Standard for the Transmission of IP Datagrams over FDDI Networks* D. Katz

1191 *Path MTU Discovery* J. Mogul, S. Deering

1198 *FYI on the X Window System* R.W. Scheifler

1207 *FYI on Questions and Answers: Answers to Commonly Asked "Experienced Internet User" Questions* G.S. Malkin, A.N. Marine, J.K. Reynolds

1208 *Glossary of Networking Terms* O.J. Jacobsen, D.C. Lynch

1213 *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II* K. McCloghrie, M.T. Rose

1215 *Convention for Defining Traps for Use with the SNMP* M.T. Rose

1228 *SNMP-DPI Simple Network Management Protocol Distributed Program Interface* G.C. Carpenter, B. Wijnen

1229 *Extensions to the Generic-Interface MIB* K. McCloghrie

1230 *IEEE 802.4 Token Bus MIB* K. McCloghrie, R. Fox

1231 *IEEE 802.5 Token Ring MIB* K. McCloghrie, R. Fox, E. Decker

1236 *IP to X.121 Address Mapping for DDN* L. Morales, P. Hasse

1267 *A Border Gateway Protocol 3 (BGP-3)* K. Lougheed, Y. Rekhter

1268 *Application of the Border Gateway Protocol in the Internet* Y. Rekhter, P. Gross

1269 *Definitions of Managed Objects for the Border Gateway Protocol (Version 3)* S. Willis, J. Burruss

1270 *SNMP Communications Services* F. Kastenholz, ed.

1321 *The MD5 Message-Digest Algorithm* R. Rivest

1323 *TCP Extensions for High Performance* V. Jacobson, R. Braden, D. Borman

1325 *FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions* G.S. Malkin, A.N. Marine

1340 *Assigned Numbers* J.K. Reynolds, J.B. Postel

1348 *DNS NSAP RRs* B. Manning

1349 *Type of Service in the Internet Protocol Suite* P. Almquist

1350 *TFTP Protocol* K.R. Sollins

1351 *SNMP Administrative Model* J. Davin, J. Galvin, K. McCloghrie

1352 *SNMP Security Protocols* J. Galvin, K. McCloghrie, J. Davin

1353 *Definitions of Managed Objects for Administration of SNMP Parties* K. McCloghrie, J. Davin, J. Galvin

1354 *IP Forwarding Table MIB* F. Baker

1356 *Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode* A. Malis, D. Robinson, R. Ullmann

1363 *A Proposed Flow Specification* C. Partridge

1372 *Telnet Remote Flow Control Option* D. Borman, C. L. Hedrick

1374 *IP and ARP on HIPPI* J. Renwick, A. Nicholson

1381 *SNMP MIB Extension for X.25 LAPB* D. Throop, F. Baker

1382 *SNMP MIB Extension for the X.25 Packet Layer* D. Throop

1387 *RIP Version 2 Protocol Analysis* G. Malkin

1388 *RIP Version 2—Carrying Additional Information* G. Malkin

1389 *RIP Version 2 MIB Extension* G. Malkin

1390 *Transmission of IP and ARP over FDDI Networks* D. Katz

1393 *Traceroute Using an IP Option* G. Malkin

1397 *Default Route Advertisement In BGP2 And BGP3 Versions of the Border Gateway Protocol* D. Haskin

1398 *Definitions of Managed Objects for the Ethernet-Like Interface Types* F. Kastenholz

1416 *Telnet Authentication Option* D. Borman, ed.

1464 *Using the Domain Name System to Store Arbitrary String Attributes* R. Rosenbaum

1469 *IP Multicast over Token-Ring Local Area Networks* T. Pusateri

1535 *A Security Problem and Proposed Correction With Widely Deployed DNS Software* E. Gavron

1536 *Common DNS Implementation Errors and Suggested Fixes* A. Kumar, J. Postel, C. Neuman, P. Danzig, S.Miller

1537 *Common DNS Data File Configuration Errors* P. Beertema

1540 *IAB Official Protocol Standards* J.B. Postel

1571 *Telnet Environment Option Interoperability Issues* D. Borman

1572 *Telnet Environment Option* S. Alexander

1577 *Classical IP and ARP over ATM* M. Laubach

1583 *OSPF Version 2* J. Moy

1591 *Domain Name System Structure and Delegation* J. Postel

1592 *Simple Network Management Protocol Distributed Protocol Interface Version 2.0* B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

1594 *FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions* A.N. Marine, J. Reynolds, G.S. Malkin

1695 *Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2* M. Ahmed, K. Tesink

1706 *DNS NSAP Resource Records* B. Manning, R. Colella

1713 *Tools for DNS debugging* A. Romao

1723 *RIP Version 2—Carrying Additional Information* G. Malkin

1766 *Tags for the Identification of Languages* H. Alvestrand

1794 *DNS Support for Load Balancing* T. Brisco

1832 *XDR: External Data Representation Standard* R. Srinivasan

1850 *OSPF Version 2 Management Information Base* F. Baker, R. Coltun

1876 *A Means for Expressing Location Information in the Domain Name System* C. Davis, P. Vixie, T. Goodwin, I. Dickinson

1886 *DNS Extensions to support IP version 6* S. Thomson, C. Huitema

1901 *Introduction to Community-Based SNMPv2* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1902 *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1903 *Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1904 *Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1905 *Protocols Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1906 *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1907 *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1908 *Coexistence between Version 1 and Version 2 of the Internet-Standard Network Management Framework* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1912 *Common DNS Operational and Configuration Errors* D. Barr

1918 *Address Allocation for Private Internets* Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear

1928 *SOCKS Protocol Version 5* M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones

1939 *Post Office Protocol-Version 3* J. Myers, M. Rose

1981 *Path MTU Discovery for IP version 6* J. McCann, S. Deering, J. Mogul

1982 *Serial Number Arithmetic* R. Elz, R. Bush

1995 *Incremental Zone Transfer in DNS* M. Ohta

1996 *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)* P. Vixie

2010 *Operational Criteria for Root Name Servers* B. Manning, P. Vixie

2011 *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2* K. McCloghrie

2012 *SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2* K. McCloghrie

2013 *SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2* K. McCloghrie

2052 *A DNS RR for specifying the location of services (DNS SRV)* A. Gulbrandsen, P. Vixie

2065 *Domain Name System Security Extensions* D. Eastlake, C. Kaufman

2096 *IP Forwarding Table MIB* F. Baker

2104 *HMAC: Keyed-Hashing for Message Authentication* H. Krawczyk, M. Bellare, R. Canetti

2132 *DHCP Options and BOOTP Vendor Extensions* S. Alexander, R. Droms

2133 *Basic Socket Interface Extensions for IPv6* R. Gilligan, S. Thomson, J. Bound, W. Stevens

2137 *Secure Domain Name System Dynamic Update* D. Eastlake

2163 *Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM)* C. Allocchio

2168 *Resolution of Uniform Resource Identifiers using the Domain Name System* R. Daniel, M. Mealling

2178 *OSPF Version 2* J. Moy

2181 *Clarifications to the DNS Specification* R. Elz, R. Bush

2205 *Resource ReSerVation Protocol (RSVP) Version 1* R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin

2210 *The Use of RSVP with IETF Integrated Services* J. Wroclawski

2211 *Specification of the Controlled-Load Network Element Service* J. Wroclawski

2212 *Specification of Guaranteed Quality of Service* S. Shenker, C. Partridge, R. Guerin

2215 *General Characterization Parameters for Integrated Service Network Elements* S. Shenker, J. Wroclawski

2219 *Use of DNS Aliases for Network Services* M. Hamilton, R. Wright

2228 *FTP Security Extensions* M. Horowitz, S. Lunt

2230 *Key Exchange Delegation Record for the DNS* R. Atkinson

2233 *The Interfaces Group MIB Using SMIv2* K. McCloghrie, F. Kastenholz

2240 *A Legal Basis for Domain Name Allocation* O. Vaughn

2246 *The TLS Protocol Version 1.0* T. Dierks, C. Allen

2308 *Negative Caching of DNS Queries (DNS NCACHE)* M. Andrews

2317 *Classless IN-ADDR.ARPA delegation* H. Eidnes, G. de Groot, P. Vixie

2320 *Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2* M. Greene, J. Luciani, K. White, T. Kuo

2328 *OSPF Version 2* J. Moy

2345 *Domain Names and Company Name Retrieval* J. Klensin, T. Wolf, G. Oglesby

2352 *A Convention for Using Legal Names as Domain Names* O. Vaughn

2355 *TN3270 Enhancements* B. Kelly

2373 *IP Version 6 Addressing Architecture* R. Hinden, M. O'Dell, S. Deering

2374 *An IPv6 Aggregatable Global Unicast Address Format* R. Hinden, M. O'Dell, S. Deering

2375 *IPv6 Multicast Address Assignments* R. Hinden, S. Deering

2389 *Feature negotiation mechanism for the File Transfer Protocol* P. Hethmon, R. Elz

2428 *FTP Extensions for IPv6 and NATs* M. Allman, S. Ostermann, C. Metz

2460 *Internet Protocol, Version 6 (IPv6) S pecification* S. Deering, R. Hinden

2461 *Neighbor Discovery for IP Version 6 (IPv6)* T. Narten, E. Nordmark, W. Simpson

2462 *IPv6 Stateless Address Autoconfiguration* S. Thomson, T. Narten

2464 *Transmission of IPv6 Packets over Ethernet Networks* M. Crawford

2474 *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* K. Nichols, S. Blake, F. Baker, D. Black

2535 *Domain Name System Security Extensions* D. Eastlake

2539 *Storage of Diffie-Hellman Keys in the Domain Name System (DNS)* D. Eastlake

2553 *Basic Socket Interface Extensions for IPv6* R. Gilligan, S. Thomson, J. Bound, W. Stevens

2571 *An Architecture for Describing SNMP Management Frameworks* D. Harrington, R. Presuhn, B. Wijnen

2572 *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* J. Case, D. Harrington, R. Presuhn, B. Wijnen

2573 *SNMP Applications* D. Levi, P. Meyer, B. Stewart

2574 *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* U. Blumenthal, B. Wijnen

2575 *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* B. Wijnen, R. Presuhn, K. McCloghrie

2578 *Structure of Management Information Version 2 (SMIv2)* K. McCloghrie, D. Perkins, J. Schoenwaelder

2640 *Internationalization of the File Transfer Protocol* B. Curtin

2665 *Definitions of Managed Objects for the Ethernet-like Interface Types* J. Flick, J. Johnson

2672 *Non-Terminal DNS Name Redirection* M. Crawford

2710 *Multicast Listener Discovery (MLD) for IPv6*S. Deering, W. Fenner, B. Haberman

2711 *IPv6 Router Alert Option*C. Partridge, A. Jackson

2758 *Definitions of Managed Objects for Service Level Agreements Performance Monitoring* K. White

2845 *Secret Key Transaction Authentication for DNS (TSIG)* P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington

2874 *DNS Extensions to Support IPv6 Address Aggregation and Renumbering* M. Crawford, C. Huitema

2941 *Telnet Authentication Option* T. Ts'o, ed., J. Altman

2942 *Telnet Authentication: Kerberos Version 5* T. Ts'o

2946 *Telnet Data Encryption Option* T. Ts'o

2952 *Telnet Encryption: DES 64 bit Cipher Feedback* T. Ts'o

2953 *Telnet Encryption: DES 64 bit Output Feedback* T. Ts'o, ed.

3060 *Policy Core Information Model—Version 1 Specification* B. Moore, E. Ellesson, J. Strassner, A. Westerinen

Draft RFCs

Several areas of IPv6 implementation include elements of the following draft RFCs and are subject to change during the RFC review process.

Advanced Sockets API for IPv6
W. Richard Stevens, Matt Thomas, Erik Nordmark, Tatuya Jinmei

Basic Socket Interface Extensions for IPv6
R.E. Gilligan, S. Thomson, J. Bound, J. McCann, W. R. Stevens

Default Address Selection for IPv6
R. Draves

Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
A. Conta, S. Deering

IP Version 6 Addressing Architecture
R. Hinden, S. Deering

Appendix 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 V1R4 are complete except for the updates contained in the information APARs that may be issued after V1R4 documents went to press.
2. Information APARs are predefined for z/OS V1R4 Communications Server and may not contain updates.
3. Information APARs for OS/390 documents are in the document called *OS/390 DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IDDOCMST/CCONTENTS.
4. Information APARs for z/OS documents are in the document called *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com:80/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS.

Information APARs for IP documents

Table 2 lists information APARs for IP documents.

Table 2. IP information APARs

Title	z/OS CS V1R4	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8
IP API Guide	ii13255	ii12861	ii12371	ii11635
IP CICS Sockets Guide	ii13257	ii12862		ii11626
IP Configuration				ii11620 ii12068 ii12353 ii12649 ii13018
IP Configuration Guide	ii13244	ii12498 ii13087	ii12362 ii12493 ii13006	
IP Configuration Reference	ii13245	ii12499	ii12363 ii12494 ii12712	
IP Diagnosis	ii13249	ii12503	ii12366 ii12495	ii11628
IP Messages Volume 1	ii13250	ii12857 ii13229	ii12367	ii11630 13230
IP Messages Volume 2	ii13251	ii12858	ii12368	ii11631
IP Messages Volume 3	ii13252	ii12859	ii12369 12990	ii11632 ii12883
IP Messages Volume 4	ii13253	ii12860		
IP Migration	ii13242	ii12497	ii12361	ii11618

© Copyright IBM Corp. 1994, 2002

Table 2. IP information APARs (continued)

Title	z/OS CS V1R4	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8
IP Network and Application Design Guide	ii13243			
IP Network Print Facility		ii12864		ii11627
IP Programmer's Reference	ii13256	ii12505		ii11634
IP and SNA Codes	ii13254	ii12504	ii12370	ii11917
IP User's Guide			ii12365 ii13060	ii11625
IP User's Guide and Commands	ii13247	ii12501	ii12365 ii13060	ii11625
IP System Admin Guide	ii13248	ii12502		
Quick Reference	ii13246	ii12500	ii12364	

Information APARs for SNA documents

Table 3 lists information APARs for SNA documents.

Table 3. SNA information APARs

Title	z/OS CS V1R4	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8
Anynet SNA over TCP/IP				ii11922
Anynet Sockets over SNA				ii11921
CSM Guide				
IP and SNA Codes	ii13254	ii12504	ii12370	ii11917
SNA Customization	ii13240	ii12872	ii12388	ii11923
SNA Diagnosis	ii13236	ii12490 ii13034`	ii12389	ii11915
SNA Messages	ii13238	ii12491	ii12382 ii12383	ii11916
SNA Network Implementation Guide	ii13234	ii12487	ii12381	ii11911
SNA Operation	ii13237	ii12489	ii12384	ii11914
SNA Migration	ii13233	ii12486	ii12386	ii11910
SNA Programming	ii13241	ii13033	ii12385	ii11920
Quick Reference	ii13246	ii12500	ii12364	ii11913
SNA Resource Definition Reference	ii13235	ii12488	ii12380 ii12567	ii11912 ii12568
SNA Resource Definition Samples				
SNA Data Areas	ii13239	ii12492	ii12387	ii11617

Other information APARs

Table 4 on page 745 lists information APARs not related to documents.

Table 4. Non-document information APARs

Content	Number
OMPROUTE	ii12026
iQDIO	ii11220
index of recomended maintenace for VTAM	ii11220
CSM for VTAM	ii12657
CSM for TCP/IP	ii12658
AHHC, MPC, and CTC	ii01501
DLUR/DLUS for z/OS V1R2	ii12986
Enterprise Extender	ii12223
Generic resources	ii10986
HPR	ii10953
MNPS	ii10370
Performance	ii11710 ii11711 ii11712

Appendix 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 it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

© Copyright IBM Corp. 1994, 2002

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

© Copyright IBM Corp. 1994, 2002

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O.Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly

tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

This product includes cryptographic software written by Eric Young.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats of unlicensed books and the z/OS Licensed Product Library (LK3T-4307), which contains BookManager and PDF formats of licensed books.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
Advanced Peer-to-Peer Networking
AFP
AD/Cycle
AIX
AIX/ESA
AnyNet
APL2
AS/400
AT
BookManager
BookMaster
CBPDO
C/370
CICS
CICS/ESA
C/MVS
Common User Access
C Set ++
CT
CUA
DATABASE 2
DatagLANce
DB2
DFSMS
DFSMSdfp
DFSMShsm
DFSMS/MVS
DPI
Domino
DRDA
eNetwork
Enterprise Systems Architecture/370
ESA/390
ESCON
eServer
ES/3090
ES/9000
ES/9370
EtherStreamer
Extended Services
FAA
Micro Channel
MVS
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
MQ
Natural
NetView
Network Station
Nways
Notes
NTune
NTuneNCP
OfficeVision/MVS
OfficeVision/VM
Open Class
OpenEdition
OS/2
OS/390
OS/400
Parallel Sysplex
Personal System/2
PR/SM
PROFS
PS/2
RACF
Resource Link
Resource Measurement Facility
RETAIN
RFM
RISC System/6000
RMF
RS/6000
S/370
S/390
SAA
SecureWay
Slate
SP
SP2
SQL/DS
System/360

FFST
FFST/2
FFST/MVS
First Failure Support Technology
GDDM
Hardware Configuration Definition
IBM
IBMLink
IBMLINK
IMS
IMS/ESA
InfoPrint
Language Environment
LANStreamer
Library Reader
LPDA
MCS
System/370
System/390
SystemView
Tivoli
TURBOWAYS
UNIX System Services
Virtual Machine/Extended Architecture
VM/ESA
VM/XA
VSE/ESA
VTAM
WebSphere
XT
z/Architecture
z/OS
z/OS.e
zSeries
400
3090
3890

Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

DB2 and NetView are registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the U.S., other countries, or both.

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see http://www.intel.com/sites/corporate/tradmarx.htm .

Other company, product, and service names may be trademarks or service marks of others.

Communicating Your Comments to IBM

If you especially like or dislike anything about this document, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this document.
- If you prefer to send comments by FAX, use this number: 1-800-254-0206
- If you prefer to send comments electronically, use this network ID: usib2hpd@vnet.ibm.com

Make sure to include the following in your note:

- Title and publication number of this document
- Page number or topic to which your comment applies.

© Copyright IBM Corp. 1994, 2002

Readers' Comments — We'd Like to Hear from You

z/OS Communications Server
IP Messages:
Volume 4 (EZZ-SNM)
Version 1 Release 4

Publication No. SC31-8786-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC31-8786-02

IBM

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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

Fold and Tape | **Please do not staple** | Fold and Tape

SC31-8786-02

Cut or Fold
Along Line

Program Number: 5694–A01 and 5655–G52

Printed in U.S.A.

SC31-8786-02

Spine information:

z/OS Communications Server

z/OS V1R4.0 CS: IP Messages Volume 4 (EZZ-SNM)

Version 1
Release 4