OS/390 IBM Communications Server

IP Messages: Volume 2 (EZB)

Version 2 Release 10

SC31-8570-05

OS/390 IBM Communications Server

IP Messages: Volume 2 (EZB)

Version 2 Release 10

SC31-8570-05

Note:
Before using this information and the product it supports, be sure to read the general information under “Appendix B. Notices” on page 325.

Sixth Edition (September 2000)

This edition applies to OS/390 V2R10 (Program Number 5647-A01).

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):
1-800-227-5088

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

World Wide Web:
http://www.ibm.com/s390/os390/

IBMLink:
CIBMORCF at RALVM17

IBM Mail Exchange:
tkinlaw@us.ibm.com

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

Contents

About This Book

This book provides information about the internet protocol (IP) messages that occur in IBM Communications Server for OS/390. This book can also help you to determine whether a specific problem is a result of the OS/390 V2R10 IP implementation.

For information about how to set up, initialize, and customize your Transmission Control Protocol/Internet Protocol (TCP/IP) for Multiple Virtual Storage (MVS) system, refer to *OS/390 IBM Communications Server: IP Configuration Reference* and *OS/390 IBM Communications Server: IP Programmer's Reference*. For information about how to use the applications on your TCP/IP for MVS™ system, refer to *OS/390 IBM Communications Server: IP User's Guide*.

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

IBM Communications Server for OS/390 is an integral part of the OS/390 V2R10 family of products. For an overview and mapping of the documentation available for OS/390 V2R10, refer to the *OS/390 Information Roadmap*.

Who Should Use This Book

This book 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

Before using this book, you should be familiar with TCP/IP and the protocol commands.

Terms and Abbreviations Used in This Book

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

How the Term "internet" Is Used in This Book

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

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

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

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

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

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

The following abbreviations are also used in this book:

ES/9000®
Enterprise System/9000 processor.

ES/9370™
Enterprise System/9370 processor.

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

RPC Sun Microsystems' implementation of remote procedure calls.

SQL IBM Structured Query Language.

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

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

Term Description

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

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

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

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

The variable *hlq* is used in this book to represent the high-level qualifier of a data set name. The default high-level qualifier for your installation was set when the product was installed and customized. Because TCP/IP for MVS uses both implicit and explicit data set allocation and provides you with methods of overriding the default, this variable can have many possible values. The *OS/390 IBM Communications Server: IP Configuration Reference* and the *OS/390 IBM 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.

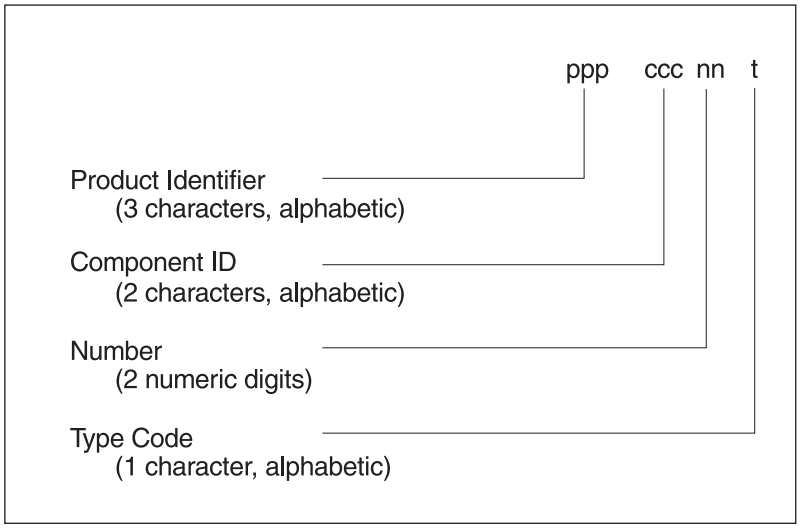

Figure 1. Sample Message Formats

The various type code characters are defined below.

Type Code
Severity

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

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

I Informational message.

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

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

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

The 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

Table 1. EZY Message Component Identifiers (continued)

EZY Component ID	Component Name
TO	Telnet Server - Other
TS	Telnet Server - State
TU	Telnet Server - Utility
TY	Telnet - System
XM	OSF/Motif
XN	OSF/Motif
XO	OSF/Motif
XP	OSF/Motif
XQ	OSF/Motif
XR	OSF/Motif
XU	OSF/Motif
XW	X Window System

Current Numbering Convention

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

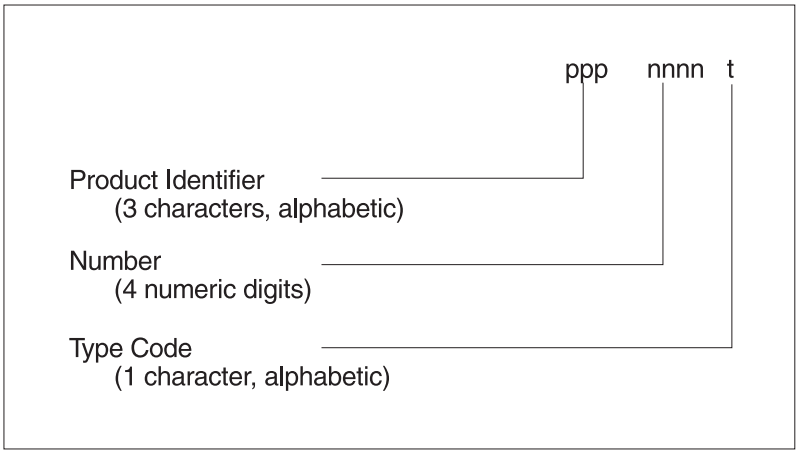

Figure 2. Sample Message Formats

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 three volumes as follows:

Volume 1
EZA Messages

Volume 2
EZB Messages

Volume 3
EZY, EZZ, and SNM Messages

Crossbook-Linking Enhancement

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

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

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

Referenced Publications

This sections list publications referenced in messages. These publications are in addition to the publications listed in the bibliography.

Referenced Publication	Order Number
CICS for MVS/ESA: User's Handbook	SX33-1188-01
OS/390® MVS Programming: Product Registration	GC28-1729-02
OS/390 MVS Programming: Authorized Assembler Services Reference ALE-DYN	GC28-1764-04
OS/390 MVS Programming: Authorized Assembler Services Reference LLA-SDV	GC28-1766-03
OS/390 MVS Programming: Workload Management Services	GC28-1773-03

Where to Find More Information

The bibliography at the end of this book describes the books in the CS for OS/390 library, arranged according to task. The bibliography also lists the files and order numbers of books related to this book, or cited by name in this book.

Most licensed books were declassified in OS/390 V2R4 and are now included in the OS/390 Online Library Collection, SK2T-6700. The remaining licensed books appear in unencrypted BookManager softcopy and PDF form on the OS/390 Licensed Product Library, LK2T-2499.

Where to Find Related Information on the Internet

You might find the following information helpful.

You can read more about VTAM, TCP/IP, OS/390, and IBM on these Web pages. For up-to-date information about Web addresses, please refer to informational APAR II11334.

Home Page	Web address
IBM Communications Server product	http://www.software.ibm.com/network/commserver/
IBM Communications Server support	http://www.software.ibm.com/network/commserver/support/
OS/390	http://www.ibm.com/s390/os390/
OS/390 Internet Library	http://www.ibm.com/s390/os390/bkserv/
IBM Systems Center publications	http://www.redbooks.ibm.com/
IBM Systems Center flashes	http://www-1.ibm.com/support/techdocs/atsmastr.nsf
VTAM and TCP/IP	http://www.software.ibm.com/network/commserver/about/csos390.html
IBM	http://www.ibm.com

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

http://www.networking.ibm.com/nsg/nsgmain.htm

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.

How to Contact IBM Service

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

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

Summary of Changes

Summary of Changes
for SC31-8570-05
IBM Communications Server for OS/390 V2R10

This edition of the book contains new and changed information that is indicated by vertical lines in the left margin.

New Information

- Added the new message EZB0942I
- Added a note to messages that reference ERRNOs or ERRNOJRs to make related information easier to find.
- Added “Chapter 10. EZB8500—EZB9744” on page 321 for messages EZB8500—EZB9744, which can be issued as either EZA or EZB.

Changed Information

- The term SecureWay has been removed from our product name. The new product name is IBM Communications Server for OS/390.

Summary of Changes
for SC31-8570-04
SecureWay Communications Server for OS/390 V2R8

The softcopy version of this book was refreshed to enable LookAt, the new online message help tool. There were no changes to the information contained in the book.

Summary of Changes
for SC31-8570-03
SecureWay Communications Server for OS/390 V2R8

This edition of the book contains new and changed information that is indicated by vertical lines in the left margin.

Changed Information

- The term eNetwork is replaced by SecureWay as part of our product name. The new name is SecureWay Communications Server.
- The name of this book has been changed to include the message prefixes. The new name of this book is: OS/390 IBM Communications Server: IP Messages Volume 2 (EZB).
- The Bibliography has been revised to show book number dash levels and delivery format.

Chapter 1. EZB0600—EZB0854

Line Printer Daemon (LPD) Messages

This section contains LPD messages. Each message will be preceded by a time stamp. When tracing is on, the date will be posted at the beginning of the trace.

EZB0600I *message*

Explanation: This message is received by the server when the dynamic allocation of a data set or the MVS enqueuing (ENQ) function was unsuccessful.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: S99Error

EZB0603I *offset*

Explanation: This message indicates the offset at which the data string buffer is initialized.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DumpString

EZB0606I *buffer position*

Explanation: This message indicates the position at which the data string buffer is initialized.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DumpString

EZB0611E Data Set name *"arguments"* **invalid.**

Explanation: An argument declared in the *hlq*.LPD.CONFIG data set is incorrect. This message indicates the incorrect argument as declared by the user.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the argument declared in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for information about the syntax rules for the *hlq*.LPD.CONFIG data set.

Source Data Set: LPD

Procedure Name: ProcessOperands

EZB0612E Data Set *"arguments"* not found.

Explanation: The argument declared in the *hlq*.LPD.CONFIG data set was not found. This message indicates the argument as declared by the user.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the argument declared in the *hlq*.LPD.CONFIG data set and restart the program. Refer to*OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessOperands

EZB0613E Data Set *"dataset name"* does not contain member *"member name"*.

Explanation: The data set name specified in the *data set name* parameter of the *hlq*.SEZAINST(LPSPROC) data set does not contain the indicated member.

System Action: LPD ends.

User or Operator Response: None.

System Programmer Response: Specify the correct data set name in the *data set name* parameter of the *hlq*.SEZAINST(LPSPROC) data set. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessOperands

EZB0614I IBM MVS LPD version *version level*

Explanation: This message indicates the current version and level of LPD for MVS.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessVersionOption

EZB0615E The option *"option"* is ambiguous. Use a longer abbreviation.

Explanation: An option specified in a parameter of the *hlq*.SEZAINST(LPSPROC) data set is incomplete. The option was abbreviated; however, the abbreviation is too short to distinguish between 2 correct options.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct option in the *hlq*.SEZAINST(LPSPROC) data set and restart the procedure. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessOptions

EZB0616E The option *"option"* was not recognized.

Explanation: An option specified in a parameter of the *hlq*.SEZAINST(LPSPROC) data set is not recognized.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct option parameter of the *hlq*.SEZAINST(LPSPROC) data set and verify that the option specified is supported on your system. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessOptions

EZB0617E Use the TRACE, TYPE or VERSION options as needed.

Explanation: An incorrect option was specified in a parameter of the *hlq*.SEZAINST(LPSPROC) data set. This message indicates the correct options for this parameter. The following list provides the names and descriptions for these options:

VERSION Displays the version number.

TYPE Activates high-lever trace facility in the LPD server. Significant events, such as the receipt of a job for printing, are recorded in the SYSOUT data set specified in your LPSPROC data set.

TRACE Causes a detailed trace of activities within the LPD server to record in the SYSOUT data set specified in your LPSPROC data set. The detailed tracing can be also activated by the DEBUG statement in the configuration data set (*hlq*.LPD.CONFIG) and by the TRACE command of the SMSG interface.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Enter the correct option and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessOptions

EZB0619E Data set prefix not specified

Explanation: The prefix name of the TRACE *name* parameter of the *hlq*.SEZAINST(LPSPROC) data set was not specified. The parameter specifies the prefix of the configuration data set. The default is *hlq*. for the LPD.CONFIG data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct data set prefix in the PREFIX *name* parameter of the *hlq*.SEZAINST(LPSPROC) data set. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessOptions

EZB0620E Program error: Invalid option *rc*

Explanation: An error has occurred during processing of the configuration data set. This message indicates the return code received for this procedure.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Check that the correct option parameters have been declared in the *hlq*.SEZAINST(LPSPROC) data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information. If the error persists, contact the IBM Software Support Center.

Source Data Set: LPD

Procedure Name: ProcessOptions

EZB0621I LPD starting with port *port number*

Explanation: This message indicates the port number on which the LPD connection was initiated.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessArguments

EZB0622E *message*

Explanation: The procedure Restore, which resets the LPD counters to the default values and performs a cleanup routine, was initiated. This message indicates the reason for the procedure call.

System Action: LPD ends.

User or Operator Response: None.

System Programmer Response: Check the message content for an indication of configuration problems. Reinitiate TCPIP, if required.

Source Data Set: LPD

Procedure Name: Restore.

EZB0623I *SayCalRe*

Explanation: This message indicates the return code received by the Restore procedure. This procedure initiates a clean-up routine and ends TCPIP processing. This message is displayed with EZB0622I.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD ends.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: Restore

EZB0625E Out of storage for connections!

Explanation: There is not enough storage allocated for a connection to complete.

System Action: The program ends abnormally.

User or Operator Response: None.

System Programmer Response: Allocate more storage for connections.

Source Data Set: LPD

Procedure Name: AllocConnection

EZB0626I Allocated ConnectionBlock at *address*

Explanation: A connection block was allocated at the indicated IP address. A connection block is used to build a connection record.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: AllocConnection

EZB0627I **Passive open on port** *port number*

Explanation: A passive connection opening was established on the indicated port number.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetNewConnection

EZB0628I **Allocated PrinterBlock at** *address*

Explanation: A printer block was allocated at the indicated IP address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: AllocPrinter

EZB0629I *printer name* **added.**

Explanation: This message indicates the name of the printer that was allocated for the LPD server.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: AllocPrinter

EZB0630I **Spool allocate RC** *rc*, **Class** *default class*, **DEST** *NJEdest*, **ID** *NJE ID*, **OUTPUT** *default spool*

EZB0631I *string*

Explanation: The Network Job Entry system (NJE) is being used as the remote printing application for LPD. This message indicates the SYSOUT class, the name of the NJE node, the device user ID, the output name, and the value of the data buffer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0632I *string*

Explanation: This message indicates the size of the data buffer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0633I Use DEST and IDENTIFIER for MVS.

Explanation: The DEST and IDENTIFIER parameters should be used for this MVS system. The DEST option sets the destination node. The default is the local node. The IDENTIFIER option specifies the device user ID. The default is SYSTEM.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0634E Cannot use TAG with DEST, IDENTIFIER, OTHERS, or PRIORITY. Specify all in TAG.

Explanation: The TAG parameter can not be used with the indicated parameters. All must be specified when using the TAG parameter.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the SERVICE statement with the correct non-conflicting parameters. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0635I Spool allocate RC *spool rc*, **Class** *default class*, **OUTPUT** *default spool*

EZB0636I *string*

Explanation: The Network Job Entry system (NJE) is being used as the remote printing application for LPD. This message indicates the SYSOUT class, the name of the NJE node, the device user ID, the output name, and the value of the data buffer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0637I *string*

Explanation: This message indicates the size of the data buffer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0640E The errors reported above prevent startup of *dataset name*

Explanation: The indicated data set name could not be opened. See the previous messages, which provide more specific information about this error.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: This message is displayed with more specific error messages. Respond as indicated by the previous messages.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0641I Service *printer name* **defined with address** *address*

Explanation: This message indicates the service name and the address space that was declared in the *name* parameter of the SERVICE statement of the *hlq*.LPD.CONFIG data set. The *name* parameter specifies a service where connections are accepted and acknowledged.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: CompletePrinter

EZB0642E Use a class which is one letter.

Explanation: The CLASS=*class* parameter of the service defined in the SERVICE statement of the *hlq*.LPD.CONFIG data set is incorrect. This parameter specifies the SYSOUT class. The default is A for printers and B for punches.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct class in the CLASS=*class* parameter and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0643E Use a class which is one letter or digit.

Explanation: The CLASS=*class* parameter of the service defined in the SERVICE statement of the *hlq*.LPD.CONFIG data set is incorrect. This parameter specifies the SYSOUT class. The default is A for printers and B for punches.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct class in the CLASS=*class* parameter and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0644E Use a number after the PRIORITY keyword.

Explanation: An unexpected character was received after the PRIORITY=*priority* parameter of the service defined in the SERVICE statement of the *hlq*.LPD.CONFIG data set. The PRIORITY parameter specifies the transmission priority. The default is 50.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the entry declared in the PRIORITY=*priority* parameter and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRCSCDefault

EZB0645E SPOOL only valid for PRINTER or PUNCH

Explanation: SPOOL is valid for the PRINTER or PUNCH services only. These services are specified in the SERVICE statement of the *hlq*.LPD.CONFIG data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters declared for the services specified in the SERVICE statement. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0646E Could not spool *dataset name device***. Return code** *rc*

Explanation: The attempt to spool to the data set defined in the *name* parameter of the SERVICE statement was unsuccessful. This message indicates the return code received after this procedure.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct data set name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0647E Only use CLASS, DEST, IDENTIFIER, OTHERS, PRIORITY, SPOOL or TAG as qualifiers.

Explanation: An incorrect parameter was specified in the RSCS statement of the *hlq*.LPD.CONFIG data set. The following list provides a description for these parameters:

CLASS The SYSOUT class. The default is A for printers and B for punches.

DEST Specifies the destination node ID. The default is the node on which LPSERVE is running.

IDENTIFIER Specifies the device user ID. The default is SYSTEM.

OTHERS The option is ignored by MVS LPSERVE.

PRIORITY Specifies the transmission priority. The default is 50.

SPOOL Supplies the operands for the CP SPOOL command that will be used on the virtual printer or punch that is defined for each new job.

TAG Supplies the operands for the CP TAG command that will used on the virtual printer or punch that is defined for each new job.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct parameter in the RSCS statement of the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0648E Use a class which is one letter.

Explanation: The CLASS option of the NJE parameter for the SERVICE statement in the *hlq*.LPD.CONFIG data set is not correct. The default is A for printers and B for punches.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct CLASS option for the NJE parameter in the SERVICE statement and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0649E Use a class which is one letter or digit.

Explanation: The CLASS option of the NJE parameter for the SERVICE statement in the *hlq*.LPD.CONFIG data set is not correct. The default is A for printers and B for punches.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct CLASS option for the NJE parameter in the SERVICE statement and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0650E SPOOL only valid for PRINTER or PUNCH

Explanation: SPOOL is valid for the PRINTER or PUNCH services only. These services are specified in the SERVICE statement of the *hlq*.LPD.CONFIG data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the parameters declared for the services specified in the SERVICE statement. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0651E Could not spool *dataset name device*. **Return code** *rc*

Explanation: The attempt to spool to the data set defined in the *name* parameter of the SERVICE statement was unsuccessful. This message indicates the return code received after this procedure.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct data set name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0652E Only use CLASS, DEST, SPOOL or TAG as qualifiers.

Explanation: An incorrect parameter was specified in the LOCAL statement of the *hlq*.LPD.CONFIG data set. The following list provides a description for these parameters:

CLASS The SYSOUT class. The default is A for printers and B for punches.

DEST Specifies the destination node ID. The default is the node on which LPSERVE is running.

SPOOL Supplies the operands for the CP SPOOL command that will be used on the virtual printer or punch that is defined for each new job.

TAG Supplies the operands for the CP TAG command that will used on the virtual printer or punch that is defined for each new job.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct parameter in the RSCS statement of the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: SetPrinterRSCSDefault

EZB0653E The keyword *"keyword"* **is not a keyword.**

Explanation: An incorrect parameter was declared in the SMTP statement of the *hlq*.LPD.CONFIG data set. The SMTP statement specifies the SMTP server name, CLASS, and DEST options.

System Action: LPD continues. Failed job will not work unless the SMTP statement is defined and LPD is restarted.

User or Operator Response: None.

System Programmer Response: Correct the parameter declared in the SMTP statement of the *hlq*.LPD.CONFIG data set, and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0654E The keyword *"keyword"* **is too short. Use a longer abbreviation.**

Explanation: The indicated option was abbreviated. However, the abbreviation is too short to distinguish between 2 correct options.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct option and restart the procedure. Refer to *OS/390 IBM*

Communications Server: IP Configuration Reference for more information.

Source Data Set: LPD

Procedure Name: ProcessLocalandRCSCOptions

EZB0655E Use the SMTP service machine name after "SMTP".

Explanation: An incorrect parameter was specified in the SMTP statement of the *hlq*.LPD.CONFIG data set. This statement specifies the SMTP server name, CLASS, and DEST options.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct name in the *server_name* parameter of the SMTP statement of the *hlq*.LPD.CONFIG data set and restart the program. If this parameter is omitted, the default is SMTP. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0656E The keyword *"keyword"* is not a keyword.

Explanation: An incorrect parameter was declared in the SMTP statement of the *hlq*.LPD.CONFIG data set. The SMTP statement specifies the SMTP server name, CLASS, and DEST options.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the parameter declared in the SMTP statement of the *hlq*.LPD.CONFIG data set, and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0657E The keyword *"keyword"* is too short. Use a longer abbreviation.

Explanation: A parameter of the SMTP statement in the *hlq*.LPD.CONFIG data set is incorrect. This parameter was abbreviated; however, the abbreviation is too short to distinguish between 2 correct parameters.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct parameter in the SMPT statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0658E Use a class which is one letter.

Explanation: The class specified in the CLASS=*class* parameter of the SMTP statement of the *hlq*.LPD.CONFIG data set is incorrect. This parameter specifies the SYSOUT class. The default is A for printers and B for punches. Valid values for this parameter are A through Z.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the class declared in the CLASS=*class* parameter of the SMTP statement in the *hlq*.LPD.CONFIG data set and restart the program.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0659E Use a class which is one letter or digit.

Explanation: The class specified in the CLASS=*class* parameter of the SMTP statement of the *hlq*.LPD.CONFIG data set is incorrect. This parameter specifies the SYSOUT class. The default is A for printers and B for punches. Valid values for this parameter are A through Z and 0 through 9.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the class declared in the CLASS=*class* parameter of the SMTP statement in the *hlq*.LPD.CONFIG data set and restart the program.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0660E Only use CLASS, DEST, IDENTIFIER or OTHERS as qualifiers.

Explanation: A parameter of the STMP statement in the *hlq*.LPD.CONFIG data set is incorrect. This message indicates the correct parameters for this statement.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct parameter for the STMP statement of the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: ProcessSMTPOptions

EZB0665E Cannot open configuration file.

Explanation: The configuration data set could not be opened.

System Action: LPD ends.

User or Operator Response: None.

System Programmer Response: Check that the name of the configuration data set was specified correctly. Check that the data set is available to the server. Check that the server has authority to access the data set.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0666E Use *option* after you have defined a SERVICE.

Explanation: A parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set was not found. The SERVICE statement specifies a service for which connections are accepted and acknowledged. The following provides a description of the valid parameters for the SERVICE statement:

name The service name must be 1 to 8 characters in length. Only characters permitted in MVS data set names are valid. This value is case-sensitive.

PRINTER Specifies that the service is to a printer.

PUNCH Specifies that the service is to a punch device.

NONE Specifies that the service is not currently in use.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct parameter for in the SERVICE statement of the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0667E The keyword *"keyword"* is ambiguous. Use a longer abbreviation.

Explanation: A statement of the *hlq*.LPD.CONFIG data set is incorrect. An abbreviation was used for this statement; however, the abbreviation is too short to distinguish between 2 correct statements.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Use the correct statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0668E The keyword *"keyword"* was not recognized.

Explanation: A statement in the *hlq*.LPD.CONFIG data set is incorrect.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the statement in the *hlq*.LPD.CONFIG data set and restart the job. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0669E Use an integer after "DISK or UNIT".

Explanation: A number is expected after the DISK or UNIT statements of the *hlq*.LPD.CONFIG data set.

System Action: LPD halts.

User or Operator Response: None.

System Programmer Response: Specify the correct value in the parameters of the DISK or UNIT statement of the *hlq*.LPD.CONFIG data set and restart the program.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0670E Use either START or END after "EXIT".

Explanation: Specify the START or END parameter with the EXIT statement of the *hlq*.LPD.CONFIG data set. The following provides a description for these parameters:

Parameter	Description
START	Specifies that the program is invoked after allocating and opening the output data set, but before anything is written to the data set.
END	Specifies that the program is invoked just before closing the output data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Use the correct parameter with the EXIT statement of the *hlq*.LPD.CONFIG data

set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0671E Use either START or END after "EXIT".

Explanation: An incorrect parameter was specified in the EXIT statement of the *hlq*.LPD.CONFIG data set. The following provides a description for these parameters:

Parameter	Description
START	Specifies that the program is invoked after allocating and opening the output data set, but before anything is written to the data set.
END	Specifies that the program is invoked just before closing the output data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Use the correct parameter with the EXIT statement of the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0672E Use name after type of EXIT.

Explanation: The program name specified in the *program* parameter of the EXIT statement in the *hlq*.LPD.CONFIG data set is incorrect or not found. This parameter specifies the name of the program to be invoked after allocating and opening, but before closing, an output data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct program name in the *program* parameter of the EXIT statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0673E Could not load EXIT *program name***.**

Explanation: The program name specified in the *program* parameter of the EXIT statement in the *hlq*.LPD.CONFIG data set could not be accessed. This parameter specifies the name of the program to be invoked after allocating and opening, but before closing, an output data set. This message is displayed when the START parameter was specified with the EXIT statement.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct program name was specified in the *program* parameter of the EXIT statement in the *hlq*.LPD.CONFIG data set and restart the program.

The library containing the program should be in the system's link list, (LNKLSTxx) or a STEPLIB definition can be used if the library is APF authorized.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0674E Could not load EXIT *program name***.**

Explanation: The program name specified in the *program* parameter of the EXIT statement in the *hlq*.LPD.CONFIG data set could not be accessed. This parameter specifies the name of the program to be invoked after allocating and opening, but before closing, an output data set. This message is displayed when the END parameter was specified with the EXIT statement.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct program name was specified in the *program* parameter of the EXIT statement in the *hlq*.LPD.CONFIG data set and restart the program.

The library containing the program should be in the system's link list, (LNKLSTxx) or a STEPLIB definition can be used if the library is APF authorized.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0675E Use either MAIL or DISCARD after "FAILEDJOB".

Explanation: A parameter specified in the FAILEDJOB statement of the *hlq*.LPD.CONFIG data is incorrect or not found. This statement specifies whether a notice of unsuccessful jobs should be mailed to users or a job is discarded without notification. The following provides a description for these parameters:

MAIL Specifies that notices of unsuccessful jobs are mailed to users.

DISCARD Specifies that unsuccessful jobs are discarded without notice.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct parameter in the FAILEDJOB statement of the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0676E Use an integer after "LINESIZE".

Explanation: The value specified in the *length* parameter of the LINESIZE statement in the *hlq*.LPD.CONFIG data set was not found. This parameter specifies the number of characters in a line on a page. Lines longer than this number are truncated. The default is 132.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the value declared in the *length* parameter of the LINESIZE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0677E Use an integer after "LINESIZE"

Explanation: The value specified in the *length* parameter of the LINESIZE statement in the *hlq*.LPD.CONFIG data set is incorrect. This parameter specifies the number of characters in a line on a page. Lines longer than this number are truncated. The default is 132.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the value declared in the *length* parameter of the LINESIZE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0678E Use only one of LOCAL, NJE, RSCS, and REMOTE.

Explanation: The destination type parameter for the SERVICE statement in the *hlq*.LPD.CONFIG data set is not correct. The following list provides a description of the valid destination types for this parameter:

LOCAL Specifies that the data sets are written to the local MVS printer or punch.

NJE Specifies that the data sets are delivered to the Network Job Entry (NJE) system.

RSCS Specifies that the data sets are delivered to the Remote Spooling Communications Subsystem (RSCS).

REMOTE Specifies that the data sets are forwarded to a remote printer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the destination type parameter for the SERVICE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0679I Allocated ObeyBlock at *address*

Explanation: The OBEY statement for the *hlq*.LPD.CONFIG data set has been received. This statement specifies user IDs authorized to use the SMSG interface provided with LPD. This message indicates that an address has been allocated an ObeyBlock, or authorization to use the SMSG interface. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0680E Use an integer after "PAGESIZE".

Explanation: The value specified in the *lines* parameter of the PAGESIZE statement in the *hlq*.LPD.CONFIG data set was not found. This parameter specifies the number of lines on a page. The default is 60.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct value in the *lines* parameter of the PAGESIZE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0681E Use an integer after "PAGESIZE".

Explanation: The value specified in the *lines* parameter of the PAGESIZE statement in the *hlq*.LPD.CONFIG data set is incorrect. This parameter specifies the number of lines on a page. The default is 60.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct value in the *lines* parameter of the PAGESIZE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0682E Use a printer@hostname after "REMOTE".

Explanation: The destination specified in the *printer@hostname* parameter of the REMOTE statement in the *hlq*.LPD.CONFIG data set was not found. This parameter indicates the destination printer at a specified IP host. This can be an IP name or an IP address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct entry in the *printer@hostname* parameter of the REMOTE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0683E Use a printer@hostname after "REMOTE".

Explanation: The destination specified in the *printer@hostname* parameter of the REMOTE statement in the *hlq*.LPD.CONFIG data set is incorrect. This parameter indicates the destination printer at a specified IP host. This can be an IP name or an IP address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct entry in the *printer@hostname* parameter of the REMOTE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0684E Cannot reach *destination*

Explanation: The destination specified in the *printer@hostname* parameter of the REMOTE statement in the *hlq*.LPD.CONFIG data set could not be reached. This parameter indicates the destination printer at a specified IP host. This can be an IP name or an IP address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct destination in the *printer@hostname* parameter of the REMOTE statement, and for the correct service name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set, and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0685E Use only one of LOCAL, NJE, RSCS, and REMOTE.

Explanation: The destination type parameter specified for the SERVICE statement in the *hlq*.LPD.CONFIG data set is not correct. The following list provides a description of the valid service names for this parameter:

LOCAL Specifies that the data sets are written to the local MVS printer or punch.

NJE Specifies that the data sets are delivered to the Network Job Entry (NJE) system.

RSCS Specifies that the data sets are delivered to the Remote Spooling Communications Subsystem (RSCS).

REMOTE Specifies that the data sets are forwarded to a remote printer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the destination type parameter declared for the SERVICE statement in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0686I Host *"host"* resolved to *address*. Printer name is *"printer name"*.

Explanation: This message indicates the host name, IP address, and the printer name to which the destination address was resolved. The destination address is specified in the *printer@hostname* parameter of the REMOTE statement in the *hlq*.LPD.CONFIG data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0687E Use only one of LOCAL, NJE, RSCS, and REMOTE.

Explanation: The service name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set is incorrect. The following list provides a description of the valid service names for this parameter:

LOCAL
Specifies that the data sets are written to the local MVS printer or punch.

NJE Specifies that the data sets are delivered to the Network Job Entry (NJE) system.

RSCS Specifies that the data sets are delivered to the Remote Spooling Communications Subsystem (RSCS).

REMOTE
Specifies that the data sets are forwarded to a remote printer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the service name declared in the *name* parameter of the SERVICE parameter in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0688E Use a printer name and type after "SERVICE".

Explanation: The service name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set was not found. The SERVICE statement specifies the service name for which connections are accepted and acknowledged.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct name specified in the *name* parameter of the SERVICE statement and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference*for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0689E The service *"service name"* has been described more than once.

Explanation: The service name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set is a duplicate. This statement specifies a service for which connections are accepted and acknowledged.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct name specified in the *name* parameter of the SERVICE statement and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0690E Use a printer name and type after "SERVICE".

Explanation: The service name specified in the *name* parameter of the SERVICE statement in the *hlq*.LPD.CONFIG data set was not found. This statement specifies a service for which connections are accepted and acknowledged.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct name specified in the *name* parameter of the SERVICE statement and restart the program.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0691E Use "PRINTER" or "PUNCH" as a SERVICE type.

Explanation: The parameter specified in the SERVICE statement of the *hlq*.LPD.CONFIG data set is incorrect. This statement indicates a service for which connections are accepted and acknowledged. The following provides a description of the valid parameters for this statement:

name The service name must be 1 to 8 characters in length. Only characters permitted in MVS data set names are valid. This value is case-sensitive.

PRINTER Specifies that the service is to a printer.

PUNCH Specifies that the service is to a punch device.

NONE Specifies that the service is not currently in use.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct parameter is specified in the SERVICE statement and

restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters.

EZB0692E Use a Volume Serial after "VOLUME".

Explanation: An incorrect value was declared in the VOLUME statement of the *hlq*.LPD.CONFIG data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct entry specified in the VOLUME statement of the *hlq*.LPD.CONFIG data set and restart the program. The correct length of this parameter is 6 characters.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0693E Use a Table Name after "TRANSLATETABLE".

Explanation: The name specified in the *name* parameter of the TRANSLATETABLE statement in the *hlq*.LPD.CONFIG data set was not found. This parameter specifies the name of the translation table. If a DBCS conversion parameter is specified, *name* is used to determine which DBCS table to load.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct name was specified in the *name* parameter of the TRANSLATETABLE statement and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0694E Could not load Translate Table.

Explanation: The translation table name specified in the *name* parameter of the TRANSLATETABLE statement in the *hlq*.LPD.CONFIG data set could not be loaded. This statement specifies the translation table to be used by the client and is found in the *name*.TCPXLBIN data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct translate table name is specified in the *name* parameter of the TRANSLATETABLE statement, check that the *name*.TCPXLBIN data set is available to the server, and restart the program. If a DBCS conversion parameter is specified in the SMTP.SMTP.CONFIG data set, *name* is used to determine which DBCS translation table to load.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0695E Could not load Translate Table.

Explanation: The translation table name specified in the *name* parameter of the TRANSLATETABLE statement in the *hlq*.LPD.CONFIG data set could not be initiated. This statement specifies the translation table to be used by the client and is found in the *name*.TCPXLBIN data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct translation table name is specified in the *name* parameter of the TRANSLATETABLE statement, check that the *name*.TCPXLBIN data set is available to the server, and restart

the program. If a DBCS conversion parameter is specified in the SMTP.SMTP.CONFIG data set, *name* is used to determine which DBCS translation table to load.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0696E Program error: Invalid option *option*

Explanation: An incorrect statement was specified in the *hlq*.LPD.CONFIG data set. This message indicates the statement as declared in this data set.

System Action: TCPIP ends.

User or Operator Response: None.

System Programmer Response: Restart TCPIP, correct the statement specified in the *hlq*.LPD.CONFIG data set and restart the program. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0697I ...End of Printer chain...

Explanation: The *hlq*.LPD.CONFIG data set statements have been processed to build the control tables representing the supported printers.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0698E InitEmulation failed

Explanation: The procedure InitEmulation, which starts the process to allow commands to run in non-EC mode machines, was not successful.

System Action: TCPIP ends.

User or Operator Response: None.

System Programmer Response: Run a 3270 type terminal emulator or use a 3270 type display station and restart TCPIP.

Source Data Set: LPD

Procedure Name: PrepareTCP

EZB0699I Starting TCP/IP service connection

Explanation: A connection was initiated to the TCPIP services.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrepareTCP

EZB0700E BeginTcpip: *SayCalRe*

Explanation: The procedure BeginTcpip, which informs the TCPIP address space that you want to start using its services, was unsuccessful. This message indicates the return code received from this procedure.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: TCPIP ends.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: PrepareTCP

EZB0701I TCP/IP turned on.

EZB0702I Host *"host ID"* **Domain** *"domain ID"* **TCPIP Service Machine** *"service ID"*

Explanation: TCPIP services have been initiated for the indicated host, domain, and printer IDs.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrepareTCP

EZB0703E FSEND failed *SayCalRe*

Explanation: The procedure FSEND, which sends data on a TCP connection, was unsuccessful. This message indicates the return code received from this procedure.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: SendACK

EZB0704E Abort issued for connection *connection number*

Explanation: The procedure DoAbortConnection, which shuts down a specific connection, was initiated. This message indicates the connection number for which the TcpAbort procedure was started.

System Action: The TCP connection ends.

User or Operator Response: None.

System Programmer Response: Reinitiate the connection if required.

Source Data Set: LPD

Procedure Name: DoAbortConnection

EZB0705I *date time*

Explanation: This message indicates the current date and time.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrintTimeStamp

EZB0706E Terminating connection *connection message*

Explanation: Because of an incorrect return code received from TCP, the connection ends. This message indicates the connection number and the reason for the termination. This message is displayed with EZB0705I.

System Action: This connection ends.

User or Operator Response: None.

System Programmer Response: Reinitiate the connection if required.

Source Data Set: LPD

Procedure Name: TerminateConnection

EZB0707I Adding *"message line"* **to message.**

Explanation: The indicated message line was added to the message input buffer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoSendQueueList

EZB0708I FSend of response sent

Explanation: The procedure FSend, which sends data on a TCP connection, was initiated.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoSendQueueList

EZB0710I New command *command code* **(no operands)**

Explanation: This message indicates the command code that was received from the client.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewCommand

EZB0711I New command *command code* **data** *"parameter"***.**

Explanation: This message indicates the command code and the additional operands that have been received from the client.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewCommand

EZB0712E Command rejected. Printer *"printer"* **not recognized.**

Explanation: A command from the client was received that contains an operand with an unrecognized printer ID.

System Action: The LPD to LPD connection ends.

User or Operator Response: Reinitiate the connection, check for the correct printer ID, and resubmit the command. Refer to for more information.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: DoNewCommand

EZB0713E Printer *"print"* **not found**

Explanation: The printer specified in the remote printing command is incorrect. This message indicates the printer ID as declared by the user.

System Action: LPD continues.

User or Operator Response: Check for the correct printer ID and resubmit the command. See the *OS/390 IBM Communications Server: IP User's Guide* for more information.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: DoNewCommand

EZB0716I Job *job ID comment printer name site*

Explanation: This message indicates the job ID that was placed onto the queue for the designated printer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrintJobLogLine

EZB0717E Could not erase *dataset name* **RC=***rc*

Explanation: The job submitted by the client could not be erased from the queue after processing. This message indicates the data set name and return code that was passed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for error messages in the LPD log and trace.

Source Data Set: LPD

Procedure Name: EraseFile

EZB0718E Could not open "*queue name* QUEUE".

Explanation: The specified queue could not be opened.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: This message should be preceded by more specific messages. Correct the errors indicated by the preceding messages.

Source Data Set: LPD

Procedure Name: SavePrinterQueue

EZB0719I Allocated JobBlock at *address*

Explanation: The print job block received from the client was allocated at the indicated IP address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: AllocJob

EZB0720E Site file "*name* SITE" record 1 unreadable.

Explanation: The first record of the *hlq*.HOSTS.SITEINFO data set could not be read.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the *hlq*.HOSTS.SITEINFO data set was generated and installed, and that the records have been entered using the correct record format. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: LoadSite

EZB0721E Site file "*name* SITE" record 2 unreadable.

Explanation: The second record of the *hlq*.HOSTS.SITEINFO data set could not be read.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the *hlq*.HOSTS.SITEINFO data set was generated and installed and that the records have been entered using the correct record format.

Source Data Set: LPD

Procedure Name: LoadSite

EZB0722E Could not open "*name* SITE".

Explanation: The *hlq*.HOSTS.SITEINFO data set could not be opened. When making changes to the *hlq*.HOSTS.LOCAL data sets, you must generate and install new *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO data sets. Use the MAKESITE statement as either a TSO command or a batch job to generate the new data sets.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the *hlq*.HOSTS.SITEINFO data set was generated and installed and that the records have been entered using the correct record format.

Source Data Set: LPD

Procedure Name: LoadSite

EZB0723I Allocated StepBlock at *address*

Explanation: A StepBlock was allocated at the indicated IP address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: AllocStep

EZB0724E Could not open control file "*site name job ID printer name***". Job abandoned.**

Explanation: The control data set specified in the LPR command received by the server could not be opened. This message indicates the site, the job ID, and the remote printer ID.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check that the correct data set name was specified in the LPR command and that the data set is available to the server and reissue the LPR command.

Source Data Set: LPD

Procedure Name: ProcessControlFile

EZB0725I Reloading job *job***.**

Explanation: The job ID, specified in the remote printing command, is reloaded.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LoadJob

EZB0726E Could not open "*site job ID***".**

Explanation: The job ID specified in the LPR command received by the server could not be processed because the indicated site could not be opened. This message indicates the site and the job ID specified.

System Action: LPD continues.

User or Operator Response: Make sure the correct destination ID is used for its corresponding parameter and the correct job name is declared in the JOB parameter of the LPR command. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LoadJob

EZB0727E Job *job ID* **abandoned. Job file too short.**

Explanation: The procedure readln, which reads the data set specified by the client on the LPR command, returned a non-zero return code.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct job ID specified in the Job *jobname* parameter of the LPR command and submit the command again. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: LoadJob

EZB0728E Job *job ID* **abandoned. Job file too short.**

Explanation: The procedure readln, which reads the data set specified by the client on the LPR command, returned a non-zero return code.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct job ID specified in the Job *jobname* parameter of the LPR command and submit the command again. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: LoadJob

EZB0729E Job *job ID* **abandoned. Job file too short.**

Explanation: The procedure readln, which reads the data set specified by the client on the LPR command, returned a non-zero return code.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check for the correct job ID specified in the Job *jobname* parameter of the LPR command and submit the command again. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: LoadJob

EZB0730E Job *job ID* **abandoned. Not enough storage.**

Explanation: The job ID specified in the *jobname* parameter of the LPR command received by the server could not be completed because of insufficient storage.

System Action: LPD continues.

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

System Programmer Response: Check for the storage requirements needed to process the LPR command, allocate more storage, and reissue the command.

Source Data Set: LPD

Procedure Name: LoadJob

EZB0731I Work Queue start

Explanation: The work queue was initiated.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrintWorkQueue

EZB0732I *job number job ID*

Explanation: This message indicates the print job received from the client was placed onto the queue for the designated service.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrintWorkQueue

EZB0733I Work Queue end

Explanation: The work queue is empty. The print server returns to a passive wait state awaiting the next print request.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: PrintWorkQueue

EZB0734I Job *jobnumber:* **added to work queue**

Explanation: TCPIP displays this message while tracing is on. The specified job number is displayed as it is added to the list of queued jobs.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: QueueJobWork

EZB0735I *stepblock_address datasize action_code dataset name*

Explanation: TCPIP issues this message when tracing is on. The StepBlock address, data size, action code and data set name of the current job are displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DumpStepChain

EZB0736E Could not open *name* **QUEUE**

Explanation: LPD could not open the specified data set that is queued for printing.

System Action: LPD continues.

User or Operator Response: Check the syntax of the specified data set name or verify that you have authority to print through the system programmer.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: LoadPrinterQueue

EZB0737I Reloading *dataset name* **queue**

Explanation: LPD issues this message while tracing is on. While attempting to queue the previous data set name, an error was detected. The data set name is reloaded into the list of queued data sets.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LoadPrinterQueue

EZB0738E Ignoring job *number* **from site** *name* **because there is no SITE file.**

Explanation: TCPIP was unable to locate the site for the specified job number.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Make sure the site was defined using the MAKESITE command.

Source Data Set: LPD

Procedure Name: LoadPrinterQueue

EZB0739I Validating user *user*

Explanation: This message occurs while tracing is on. The userid of the submitter of the print job is validated.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LoadPrinterQueue

EZB0740E Validation failed. RC *rc*

Explanation: TCPIP issues this message while tracing is on. While attempting to validate the password for the currently queued job an error was detected.

System Action: LPD continues.

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

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: ValidateJob

EZB0741E Could not define *printer* **at** *address***. Return code was** *rc*

Explanation: TCPIP was unsuccessful in its attempt to recognize the specified printer and address. A return code was passed.

System Action: LPD continues.

User or Operator Response: Check the syntax and/or names of the specified printer and address. If the problem persists, contact the system programmer.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: DefineCPDevice

EZB0742I *printer name printer address* **defined**

Explanation: TCPIP issues this message while tracing is on. The specified printer and address are defined.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DefineCPDevice

EZB0744I *address punch line*

Explanation: LPD displays the address of the punch command text and the punch line number. The punch line text represents the information sent to SMTP from LPD.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: Punchline

EZB0747E Could not allocate SMTP Spool

Explanation: The SMTP device could not allocate the spool for the batch data set submitted using the SMTP command.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Make sure the SMTP device is activated.

Source Data Set: LPD

Procedure Name: SendFailingMail

EZB0748E Could not open spool to *SMTP***. Return code was** *last error*

Explanation: LPD could not spool a job to the indicated SMTP service. A non-zero return code was returned.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Make sure the SMTP statement is updated to include accurate information for its

parameters in the *hlq*.LPD.CONFIG data set. Also make sure that the MAIL parameter is used with the FAILEDJOB statement. For more information about the SMTP and FAILEDJOB statements, refer to *OS/390 IBM Communications Server: IP Configuration Reference*. Check the return code issued in this message and the return codes listed under SMTP in the *OS/390 IBM Communications Server: IP and SNA Codes* to further determine and correct the error.

Source Data Set: LPR

Procedure Name: SendFailingMail

EZB0750E Could not deallocate Spool File *ddname* **Error code was** *rc*

Explanation: The SMTP server was unable to deallocate the specified spool file after attempting to close the SMTP connection.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Check the return code issued in this message and the return codes listed under SMTP in the *OS/390 IBM Communications Server: IP and SNA Codes* to determine and correct the error.

Source Data Set: LPR

Procedure Name: SendFailingMail

EZB0751I Released StepBlock at *address*

Explanation: The SMTP server displays this message as the job at the specified address is deleted from storage.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: RemoveJobFiles

EZB0753I New subcommand *command* **(no operands)**

Explanation: This message is displayed while tracing is on. A new SMTP subcommand was detected by LPD with no optional operands.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DoNewSubcommand

EZB0754I New subcommand *command* **operands** *operands***.**

Explanation: This message is displayed while tracing is on. A new SMTP subcommand was detected by LPD. The new subcommand and optional operands are displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DoNewSubcommand

EZB0755I Released StepBlock at *address*

Explanation: This message occurs while tracing is on. This message is displayed when the previous subcommand is deleted from the stated storage area.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DoNewSubcommand

EZB0756I Job number *number* **is invalid.**

Explanation: The LPD server was unable to queue the specified job number for processing. The connection was terminated.

System Action: LPD continues.

User or Operator Response: Resubmit the job using the LPR command.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: DoNewSubcommand

EZB0757I Duplicate file name *"dataset name"***.**

Explanation: This message indicates that the specified data set was previously recognized by LPD. The job is ignored and the next job is processed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewSubcommand

EZB0759E Failed to allocate block for *jobnumber* **from** *site*

Explanation: The service machine was unable to allocate enough storage to process the specified job at the specified site.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewSubCommand

EZB0760E Failed to open *"dataset name"***.**

Explanation: The server machine's attempt to open the stated data set failed. The connection is terminated.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Make sure that the specified data set is defined in the MAKESITE tables.

Source Data Set: LPD

Procedure Name: DoNewCommand

EZB0761E Could not open data file *"dataset name"***. Job abandoned.**

Explanation: The server machine issues this message after attempting to open the data file at the specified site with the specified file type. The job is abandoned.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Make sure the specified site is defined by the MAKESITE command.

Source Data Set: LPR

Procedure Name: DoSendStep

EZB0762I Sending subcommand *command* **with an operand of** *operand*

Explanation: LPD issues this message while tracing is on. LPD acknowledges sending the indicated subcommand with the indicated operand.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DoSendJob

EZB0763I Closing connection *connection*

Explanation: This message is issued while tracing is on. The connection between the client and the LPD server has been closed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoCloseConnection

EZB0764I ACK received on connection *connection* **for job** *jobnumber* **in state** *jobstate*

Explanation: This message is issued while tracing is on. LPD has received an acknowledgment of the specified connection. The current state of the job is also displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewAck

EZB0765E ACK in unexpected job state *jobstate*

Explanation: This message is issued while tracing is on. LPD received an ACK from the remote host, but the job state was bad. See message EZB0764I for job and connection number.

System Action: LPD continues.

User or Operator Response: Check all physical ports and power switch to verify that the printer is ready for printing. If the problem persists, contact your hardware support personnel.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewAck

EZB0766E NACK has value *number*

Explanation: This message is issued while tracing is on. The remote host refused to complete processing for this job.

System Action: LPD continues.

User or Operator Response: Check the printer and make sure it is active. If the problem persists, contact your hardware support personnel.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewAck

EZB0767I Timer cleared for connection *connection*

Explanation: This message is issued while tracing is on. LPD has successfully cleared the timer for the specified connection.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewData

EZB0768I Ignoring Data delivered on connection *connection*

Explanation: This message is issued while tracing is on. The buffer has been exhausted. The data to be delivered is ignored.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Increase the specified buffer size using the DATABUFFERPOOLSIZE statement.

Source Data Set: LPD

Procedure Name: DoNewData

EZB0769I Job *jobnumber* **removed from work queue**

Explanation: This message is issued while tracing is on. LPD issues this message when the specified job number is removed from the list of queued jobs.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: RemoveJobWork

EZB0770E Job *jobnumber* **not found in printer chain.**

Explanation: LPD issues this message when the specified job number is not found in the print queue.

System Action: LPD halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: Check the LPD trace. Job may have already printed. Restart LPD.

Source Data Set: LPD

Procedure Name: RemoveJobPrinter

EZB0771I Released JobBlock at *address*

Explanation: LPD issues this message while tracing is on. The storage space allocated for the job block at the specified address was released.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: FreeJob

EZB0772I End Connection *connection* **for** *SayCalRe*

Explanation: The specified LPD server connection has ended.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0773I Connection *connection* **terminated for** *SayCalRe*

Explanation: This message indicates that the specified connection has been terminated.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0774E Connection ended abruptly.

Explanation: The LPD connection with the remote printer has terminated. LPD received a job and the connection state was not CONNCLOSING.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Review the trace and restart the job. If DEBUG is not on, define DEBUG in the LPD.CONFIG and restart the job.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0775I Released StepBlock at *address*

Explanation: This message is issued while tracing is on. The block at the specified address was freed of storage.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0776I Released StepBlock at *address*

Explanation: This message is issued while tracing is on. The storage space specified was released because there were no jobs awaiting processing.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0777I Released ConnectionBlock at *address*

Explanation: This message is issued while tracing is on. The storage space specified for the LPD connection was released because there were no jobs awaiting processing.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0778S LPD terminating because *SayCalRe*

Explanation: This message is displayed while tracing is on. The LPD server has terminated operation.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD halts.

User or Operator Response: Notify system programmer.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0779I New connection state *state* **on connection** *address* **with reason** *SayCalRe*

Explanation: This message indicates that LPD has made a new connection state at the specified address.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: DoNewConnState

EZB0780E Abort failed *SayCalRe*

Explanation: This message is issued as a result of the LPD's unsuccessful attempt to disconnect the TCP connection.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: DoEndConnection

EZB0781E Connection aborted because port number (number) is out of range.

Explanation: This message is issued while tracing is on. The LPD connection was aborted because the stated port number was not defined as a legitimate port. Legitimate ports range from 721 to 731, inclusive.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Use the PORT statement to define the legitimate ports. For more information about the PORT statement, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: LPD

Procedure Name: DoNewConnState

EZB0782I Connection open. Reading command.

Explanation: This message is issued while tracing is on. The LPD connection is open and now reading the submitted connection state command.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoNewConnState

EZB0783E Aborting Connection *connection* **- Timed out**

Explanation: The specified connection was aborted because the timer expired.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Assist the user as necessary. Use the TIMER command to set the timer to an appropriate time limit.

Source Data Set: LPD

Procedure Name: DoConnectionTimeOut

EZB0784E Could not retrieve SMSG

Explanation: LPD was unable to retrieve a queued Smsg. Therefore, the message could not be queued for processing.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Refer to message EZB0800I.

Source Data Set: LPD

Procedure Name: ProcessSMSG

EZB0785E Attempted SMSG from "*user***" ignored.**

Explanation: The SMSG from the specified user was ignored. LPD was unable to locate the userid in the OBEYFILE.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Make sure the specified user id is listed in the OBEYFILE using the OBEY command.

Source Data Set: LPD

Procedure Name: ProcessSMSG

EZB0786I Command received "*string***"**

Explanation: LPD issues this message when the specified special messages string is received.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessSMSG

EZB0787E CP command failed. RC=*rc* **Response** *response*

Explanation: LPD received a return from a CP command through the SMSG interface. The indicated response provides more information about this error.

System Action: LPD continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Use the response string provided to correct the error and reissue the CP command.

Source Data Set: LPD

Procedure Name: ProcessSMSG

EZB0788E Command not understood.

Explanation: LPD does not recognize the command issued through the SMSG interface.

System Action: LPD continues.

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

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessSMSG

EZB0789I GetNextNote with ShouldWait of *number*

Explanation: This message is issued while tracing is on. TCP is initializing processing procedures to retrieve the next queued notification. The ShouldWait function is set to either a true or false value depending on whether TCP is to wait for notification before the next one is available.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0790I GetNextNote returns. Connection *connection* **Notification** *notification*

Explanation: This message is issued while tracing is on. The connection number and the notification status are returned upon successful completion of the GetNextNote procedure.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0791I New TCP notice arrived

Explanation: This message is issued while tracing is on. TCP has received the next queued notification.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0792I Connection: *connection*

Explanation: This message is issued while tracing is on. This message displays the connection number of the TCP connection.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0793I Notification: *notification*

Explanation: This message is issued while tracing is on. This message displays the notification status of the current TCP connection.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0794I NewState: *newstate*

Explanation: This message is issued while tracing is on and notification has changed. The new state is displayed in this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0795I ConnState: *connectionstate*

Explanation: This message is issued while tracing is on. If TCP doesn't detect a connection state change, this message is displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0796I BytesDelivered: *number*

Explanation: This message is issued while tracing is on. The number of data bytes delivered on the TCP connection are displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0797I SendTurnCode: *SayCalRe*

Explanation: This message is issued while tracing is on. TCP failed in its attempt to send data on the TCP connection.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0798I Queueing job *jobnumber*

Explanation: This message is issued while tracing is on. The specified job is queued for processing.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0799I Reading additional data on *connection*

Explanation: This message is issued while tracing is on. Processing of the current job continues on the specified connection.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0800I Ignoring TCP/IP notice *notification* **on** *connection*

Explanation: This message is issued while tracing is on. The TCP connection has failed or was not recognized. Use the type identifier issued in this message and the *OS/390 IBM Communications Server: IP Programmer's Reference* to determine and correct the problem.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD

Procedure Name: ProcessTCP

EZB0801I Filter *option* not supported. Job abandoned.

Explanation: This message could display for one of the following reasons:

- The option type of the filters parameter used with the LPR command is not supported by the line printer daemon.
- The option type of the filters parameter for the SERVICE statement in the *hlq*.LPD.CONFIG data set is not supported by the line printer daemon.

System Action: LPD continues.

User or Operator Response: Correct the option type of the filters parameter for the LPR command or in the SERVICE statement of the *hlq*.LPD.CONFIG data set. Reissue the job. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* or the *OS/390 IBM Communications Server: IP User's Guide* for more information.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoStartStep

EZB0802E Could not open data file *dataset* Job abandoned.

Explanation: LPD was unable to access the indicated data set. The print job is not completed.

System Action: LPD continues.

User or Operator Response: Reissue the print job with a valid data set name or PDS member. See the *OS/390 IBM Communications Server: IP User's Guide* for more information about using the LPR command. None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoStepStart

EZB0803E Could not define device for job. Unknown type *type* treated as "NONE".

Explanation: LPD encountered a printer type defined in the SERVICE statement that was not valid or unknown.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Correct the device type parameter for the SERVICE statement in *hlq*.LPD.CONFIG data set. For more information on the SERVICE statement, refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0804E *printer* **kind is unknown** *address*

Explanation: LPD was not able to recognize the route parameter for the indicated printer. The route parameter was defined to something other than remote, local or NJE.

System Action: LPD continues.

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

System Programmer Response: Correct the route parameter for the SERVICE statement in the *hlq*.LPD.CONFIG data set. For information on the SERVICE statement, refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0805E Could not allocate Spool Class *class*

Explanation: The LPD server was unable to set a spool file for the specified printer class due to buffer exhaustion. See messages EZB0806I and EZB0807I.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Resubmit the LPD command.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0806I Copies *copies*, **Font** *font*, **form** *form*, **output** *printer*

Explanation: LPD issues this message when a non-zero return code is returned during spool file allocation. The number of copies, font size, form, and printer name are provided. See message EZB0807I.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0807I *bufferlength*

Explanation: The length of the data buffer which failed while attempting to allocate a spool file is displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0808I *bufferlength*

Explanation: The length of the alternate data buffer which failed while attempting to allocate a spool file is displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0809E Could not open spool to *destination* **Return code was** *error*

Explanation: LPD was unable to open a spool file to the specified printer destination. A return code is passed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Use the return code listed in this message and *OS/390 IBM Communications Server: IP and SNA Codes*.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0810I **Spool** *printertype address* **for** *user*. **Return code was** *rc*

Explanation: LPD issues this message while tracing is on or a non-zero return code is displayed. The printer type, address, user, and return code are displayed with this message after the spool is allocated.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Check the meaning of the RC issued by CP spool "FOR" command.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0811E *printertype* **kind is unknown** *deviceaddress*

Explanation: The LPR command has a destination defined that is not known. The destination is defined after the jobname parameter.

System Action: LPD continues.

User or Operator Response: Specify the correct destination in the jobname parameter and reissue the LPR command. For more information on the LPR command see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0812I **Spool TO** *address* **superseded by FOR** *user*

Explanation: The SMTP spool to the default user address was superseded by the specified user id.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0813I **Spooling** *printer* **this way** *how*

Explanation: This message is displayed while tracing is on. LPD acknowledges the parameter indicated for the LPR command on the specified printer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0814E **Could not spool** *printer address type*. **Return code was** *rc*

Explanation: This message is issued while tracing is on. The LPD server was not able to spool the printer at the stated address with the specified type. A non-zero return code was returned.

System Action: Printer not spooled.

User or Operator Response: None.

System Programmer Response: Use the return code value to determine the error and reissue the command.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0815I Tagging *printer* **with** *tag*

Explanation: This message is issued while tracing is on. The indicated RSCS tag has been assigned to the indicated printer.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0816E Could not TAG *printer address tag* **Return code was** *rc*

EZB0817I Response was *response*

Explanation: The LPD server was not able to assign the specified tag to the printer at the indicated address. A non-zero return code was passed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Use the return code value and the response string provided with message EZB0817I to determine and correct the cause of the error.

Source Data Set: LPD

Procedure Name: GetJobDevice

EZB0819E Job *jobnumber* **rescheduled -- no storage**

Explanation: This message is issued while tracing is on. LPD was unable to allocate a connection for the specified job number. No storage block was available for a connection.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoStartSending

EZB0820I Trying to open with local port *port*

Explanation: This message is issued while tracing is on. LPD is attempting to open the specified local port for communication.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoStartSending

EZB0821E Job *jobnumber* **abandoned -- Open failed or no ports**

Explanation: This message is issued while tracing is on.

- LPD was unable to open a connection to the specified REMOTE Print Server (LPD) on another host. A TcpOpen error occured.
- LPD was unable to open a local port to the specified REMOTE Print Server (LPD) on another host. No ports in the range 721–731 were available.

In the message text:

jobnumber The number assinged by the LPR client to this job.

System Action: LPD continues.

User or Operator Response: Resubmit the print job when a connection to the REMOTE print server can be established.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoStartSending

EZB0822I Sending command *number* **with operand** *address*

Explanation: This message is issued while tracing is on. LPD is sending the specified command to the specified printer address.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoOpenSending

EZB0823I Sending ACK at end of data file on *connection* **for job** *jobnumber*

Explanation: This message is issued while tracing is on. The LPD server has read the dataset on the specified connection for the specified job number and is sending an ACK.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: DoContinueSending

EZB0824I ProcessWork starting on job queue

Explanation: This message is issued while tracing is on. LPD is now processing the next job queued.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessWork

EZB0825I Job *jobnumber* **for** *printer* **dispatched in state** *state*

Explanation: This message is issued while tracing is on. The job number, printer, and print state are displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessWork

EZB0826E Job *jobnumber* **(state) abandoned - Incorrect state.**

Explanation: LPD has terminated the processing of the specified job number because an incorrect job state was detected.

System Action: LPD continues.

User or Operator Response: Resubmit the job using the LPR command.

System Programmer Response: Make sure the failed job is mail to the user using the MAIL command. For more information see the *OS/390 IBM Communications Server: IP Programmer's Reference*.

Source Data Set: LPD

Procedure Name: ProcessWork

EZB0827I ProcessWork end with queue

Explanation: This message is issued while tracing is on. LPD has finished processing the queued jobs.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: ProcessWork

EZB0831I IBM MVS LPD Version *version* **on** *date* **at** *time*

Explanation: When the LPD server is initialized this message is displayed. The LPD version number, date, and starting time are displayed.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LPD MAIN

EZB0833E Could not get identity!

Explanation: The LPD server was unable to identify the user id, host name, TCP service and domain name during its start up procedure. LPD does not start.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Make sure the user id, host name, TCP service, and domain name have all been specified using the LPD command and its subcommands. For more information of the LPD command and its subcommands, refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: LPD

Procedure Name: LPD MAIN

EZB0834I Ready

Explanation: LPD is initialized and ready for processing.

System Action: LPD is initialized.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LPD MAIN

EZB0835I Ignored data is *string*

Explanation: This message displays while tracing is on; it is preceded by EZB0768I. Data was delivered on the connection, but because the data is unrecognizable, it is ignored. The connection is closed.

System Action: LPD continues.

User or Operator Response: Refer to EZB0768I.

System Programmer Response: Refer to EZB0768I. Use the data from the message to determine if the LPD was sent bad data.

Source Data Set: LPD

Procedure Name: DoNewData

EZB0850I Using Table for DBCS Translate: *data set*

Explanation: The DBCS translation table has been successfully loaded from the binary translation table data set in the search order hierarchy.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPD

Procedure Name: LoadDbcsTables

EZB0851E Could not load DBCS Translate Table: *data set*

Explanation: DBCS conversion is configured for the LPD, but the required DBCS translation table could not be loaded.

System Action: The program continues.

User or Operator Response: None.

System Programmer Response: Configure a valid DBCS binary translate table data set in the search order hierarchy for the required DBCS translation table. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information about loading and customizing DBCS translation tables.

Source Data Set: LPD

Procedure Name: LoadDbcsTables

EZB0852E Use a NLS option after ″NLSTRANSLATE″.

Explanation: An NLS option is expected after the NLSTRANSLATE keyword of the SERVICE statement of the *hlq*.LPD.CONFIG data set.

System Action: LPD continues.

User or Operator Response: None.

System Programmer Response: Specify the correct value in the parameter of NLSTRANSLATE keyword of the SERVICE statement of the *hlq*.LPD.CONFIG data set and restart the program.

Source Data Set: LPD

Procedure Name: ProcessNlsOptions

EZB0853E Use an integer after *JOBPACING*.

Explanation: A number is expected after the JOBPACING statement in the hlq.LPD.CONFIG data set.

System Action: LPD continues using the default value.

User or Operator Response: None.

System Programmer Response: Specify the correct value for the parameter in the JOBPACING statement in the hlq.LPD.CONFIG data set and restart the program.

Source Data Set: LPD

Procedure Name: PreparePrinters

EZB0854E Use an integer after *STEPLIMIT*.

Explanation: A number is expected after the STEPLIMIT statement in the hlq.LPD.CONFIG data set.

System Action: LPD continues using in the default value.

User or Operator Response: None.

System Programmer Response: Specify the correct value for the parameter in the STEPLIMIT statement in the hlq.LPD.CONFIG data set and restart the program.

Source Data Set: LPD

Procedure Name: PreparePrinters

Chapter 2. EZB0900—EZB1100

Line Printer (LPT) Messages

This section contains remote printing messages.

EZB0900I *command name* **version** *version*

Explanation: Displays the line printer command used and the version level of the program. This message is displayed when you use the *version* parameter with the LPR, LPQ, LPRM, and LPRSET commands.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM, LPRSET

Procedure Name: ProcessVersionOption

EZB0901E The option *option* **is ambiguous. Use a longer abbreviation.**

Explanation: The abbreviated option submitted at the command line is ambiguous.

System Action: The command is terminated.

User or Operator Response: Reissue the desired option with a longer abbreviation at the LPR command line. For a list of valid abbreviations using the LPR command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR, LPRM

Procedure Name: ProcessOptions

EZB0902E Use a host name after HOST option.

Explanation: The HOST option was specified without indicating host name.

System Action: The command is terminated.

User or Operator Response: Reissue a valid host name at the HOST option. For a list of valid host names, contact the system operator.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0903E The option *option* **was not recognized.**

Explanation: The program cannot recognize the option you have entered. Valid options will be provided. This message will precede the valid options for your command.

System Action: The command is terminated.

User or Operator Response: Reissue the valid option using the LPR command.

System Programmer Response: None.

Source Data Set: LPQ, LPR, LPRM, LPRSET

Procedure Name: ProcessOptions

EZB0904I Use the ALL, HOST, PRINTER, TRACE, TYPE or VERSION options as needed.

Explanation: This message displays valid options that may be used with LPT, LPQ, and LPRM commands.

System Action: The command is terminated.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPQ, LPRM

Procedure Name: ProcessOptions

EZB0905E Use a printer name after PRINTER option.

Explanation: The PRINTER option was specified without specifying a printer name.

System Action: The command is terminated.

User or Operator Response: Reissue the LPR command and the PRINTER option using a valid printer. For more information about the LPR command and the PRINTER option, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0906E Program error: Invalid option *option***.**

Explanation: You specified an invalid option. To see a list of options, type HELP followed by your command or see *OS/390 IBM Communications Server: IP User's Guide* for more information.

System Action: The command is terminated.

User or Operator Response: Reissue the LPR command and a valid option.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM, LPRSET

Procedure Name: ProcessOptions

EZB0907E Cannot get a printer name.

Explanation: The program cannot find a default printer name in the *user_id*.LASTING.GLOBALV data set.

System Action: Processing continues.

User or Operator Response: Use the LPRSET command to set up a default printer name. For more information see *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPQ, LPR, LPRM

Procedure Name: ProcessOptions

EZB0908I Printer name from global variable PRINTER = "*printer***"**

Explanation: LPR displays the default printer name, taken from the *user_id*.LASTING.GLOBALV data set. For more information about the *user_id*.LASTING.GLOBALV data set, refer to *OS/390 IBM Communications Server: IP Configuration Reference*.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0909E Use the PRINTER option this way: PRINTER nameofprinter<@printerhost>

Explanation: You issued the PRINTER option with the *at-sign* without specifying the printer name.

System Action: The command is terminated.

User or Operator Response: Reissue the printer option using the correct syntax as described in the message.

System Programmer Response: None.

Source Data Set: LPR, LPRM, LPQ

Procedure Name: ProcessOptions

EZB0910E Cannot get the printer host.

Explanation: The program cannot find the default host name from the *user_ id*.LASTING.GLOBALV data set.

System Action: The command is terminated.

User or Operator Response: Specify the host name on the command line, or use the LPRSET command to set up a default host name. For more information about the LPRSET command see *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPQ, LPR, LPRM

Procedure Name: ProcessOptions

EZB0911I Host name from global variable PRTHOST = "*host* "

Explanation: If the LPR, LPQ, or LPRM command lines do not have a specified printer host, the GLOBALV variables *PRINTER* and *PRTHOST* are reviewed in the *user_id*.LASTING.GLOBALV data set and the name of the host in use is displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions.

EZB0912E The printer name is not known.

Explanation: The program does not recognize the printer name.

System Action: Processing continues.

User or Operator Response: Check the printer name at the server machine, and reissue the command with a valid printer name or use the LPRSET command to set a default printer name at the specified host. See the *OS/390 IBM Communications Server: IP User's Guide* for more information about setting up a default printer in the *user_id*.LASTING.GLOBALV data set.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0913E The host name is not known.

Explanation: The program cannot determine the host name.

System Action: The command is terminated.

User or Operator Response: Use a valid host name with the command. The LPRSET command can be used to set up a default host name for a default printer. See *OS/390 IBM Communications Server: IP User's Guide* for more information about setting up a default host in the *user_id*.LASTING.GLOBALV data set.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0914E Please specify the printer name and host either as a command option or with the LPRSET command.

Explanation: The program cannot determine the printer name or the printer host.

System Action: Processing continues.

User or Operator Response: Reissue your command with a valid printer name and host. The LPRSET command can be used to set a default printer and host using the GLOBALV variables in the *user_id*.LASTING.GLOBALV data set. For more information, see *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: If the problem continues, obtain more information using the TYPE or TRACE functions.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0915I Begin "*cmd*" to printer "*printer*" at host "*foreignhost*"

Explanation: This message indicates which printer and remote host are being used for your task. The task is indicated by "*cmd*" in the message, which can be LPR, LPQ, or LPRM.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: ProcessOptions

EZB0916I Sending command *command* argument: *operand*

Explanation: This message indicates the command was successfully sent to the remote host.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: SendCommand

EZB0917I Command successfully sent

Explanation: The command was successfully sent and acknowledged by the remote print server.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: SendCommand

EZB0918E Command was not sent successfully

Explanation: The command was not acknowledged by the remote host; therefore, the transmission was unsuccessful.

System Action: The command is terminated.

User or Operator Response: Reenter the command using the correct syntax.

System Programmer Response: Assist the operator if necessary.

Source Data Set: LPQ, LPRM

Procedure Name: SendCommand

EZB0919I InitEmulation failed

Explanation: An error occurred initializing EC mode emulation. EC mode emulation is required to run TCPIP on a non-EC mode machine.

System Action: Processing terminated.

User or Operator Response: Reinitialize EC mode to on. If a problem occurs contact your system programmer.

System Programmer Response: EC-mode emulation can only be accessed when the processor is running. If a program is running in a 24-bit addressing mode, the program can be changed to a 31-bit addressing mode, which causes a branch to a module residing above the 16Mb virtual storage. For more information see *IBM Assembler Language Programming Book*.

Source Data Set: LPQ, LPR, LPRM

Procedure Name: LPR Main

EZB0920I Requesting TCP/IP service at *date time*

Explanation: This message indicates the time and date that LPR, LPQ, or LPRM requested TCPIP service. This message is displayed if the TRACE option is specified.

System Action: Process continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0921I Granted TCP/IP service at *date time*

Explanation: TCPIP service is provided for the command requested.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0922I Resolving *foreignhost* **at** *date time*

Explanation: The remote host is being resolved at the indicated time and date.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: DrainConnection, PrintLineWithTabs

EZB0923I Both the printer and the host name are not known.

Explanation: The program could not determine the printer name or the host name.

System Action: The command is terminated.

User or Operator Response: Check the host name and printer name for the correct information and reenter the command. Use the LPRSET command to set a default printe and host name. For more information see *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPQ, LPR, LPRM

Procedure Name: ProcessOptions

EZB0924I Host *host* **name resolved to** *host_addr* **at** *date time*

Explanation: The remote host name has been resolved to the indicated internet address along with the time and date.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0925I TCP/IP turned on.

Explanation: TCPIP services are now ready for use.

System Action: Processing continues with the next request.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0926I Host "*host***" Domain "***domain***" TCP/IP Service Machine** *name*

Explanation: This message indicates the host name, its domain name equivalent, and the TCPIP service machine in use.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0927I Trying to open with local port *port* **to foreign host address** *address*

Explanation: This message is issued while tracing is on and LPR is first initiated. TCPIP is attempting to open a connection with the stated local port to the stated foreign host address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0928I Connection open from local port *port* **to foreign host address** *address*

Explanation: This message is issued while tracing is on. TCPIP indicates that the specified local port has successfully connected to the specified foreign host address.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0929I Connected to *host*

Explanation: The local port has completed a successful telecommunication line to the remote port. This means your command has been accepted and transmitted through TCPIP services.

System Action: Processing continues with the next request.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0930I Connection closed

Explanation: TCPIP services ends the connection when your task has completed processing.

System Action: Processing ends successfully.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main, LPQ Main, or LPRM Main

EZB0931I Notification: *SayNotEn*

Explanation: This message provides information about whether the data has been delivered and whether the connection state has been changed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPQ, LPRM

Procedure Name: DrainConnection, ReceiveData

EZB0932I NewState: *SayConSt*

Explanation: The state of the connection to TCPIP has changed. The new state is indicated in the message.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPQ, LPRM

Procedure Name: DrainConnection, ReceiveData

EZB0933I ConnState: *SayConSt*

Explanation: This message indicates the state of the connection to TCPIP.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPQ, LPRM

Procedure Name: DrainConnection, ReceiveData

EZB0934I BytesToRead: *bytes*

Explanation: This message indicates the number of bytes contained in a transmission from TCPIP.

System Action: The system continues processing.

User or Operator Response: If the data is not acknowledged, TCPIP will retransmit the data to make sure that the data is received.

System Programmer Response: None.

Source Data Set: LPQ, LPRM

Procedure Name: DrainConnection, ReceiveData

EZB0935I BytesDelivered: *bytes*

Explanation: This message indicates the number of bytes delivered for your task, if the trace option is used.

System Action: The system continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPQ, LPRM

Procedure Name: ReceiveBytes, ReceiveData

EZB0936E BeginTcpip: *SayCalRe*

Explanation: The function BeginTcpip, which is used to start TCPIP service for LPQ or LPRM, was unsuccessful. The reason is indicated by the *SayCalRe* portion of this message.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

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

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPQ, LPRM

Procedure Name: LPR Main, LPQ Main or LPRM Main

EZB0939E Could not get identity! (*SayCalRe*)

Explanation: TCPIP was unable to identify the user requesting service.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing ends.

User or Operator Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main, LPQ Main or LPRM Main

EZB0940E Unknown host *host*

Explanation: The specified host does not exist or is entered incorrectly.

System Action: Processing halts.

User or Operator Response: Reenter the correct host name or address using the LPRSET command.

System Programmer Response: None.

Source Data Set: LPR, LPRM, LPQ

Procedure Name: LPR Main

EZB0941E Handle: *SayCalRe*

Explanation: The Handle procedure, which specifies what notifications to receive in a given set, was unsuccessful.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

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

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0942E Connection to *host* failed.

Explanation: The attempt to connect to the specified host was not successful.

System Action: Processing halts.

User or Operator Response: Try to restart the connection by issuing the LPD command. For more information on the LPD command check the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0943I No local printer ports available now. Bind Conn failed.

Explanation: All of the local printer ports are either busy or not ready.

System Action: Processing halts.

User or Operator Response: Wait and try the command again.

System Programmer Response: None.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0944E Could not set option (*SayCalRe*)

Explanation: The procedure that sets an option for a TCP connection was unsuccessful.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPQ, LPRM LPR, LPQ, LPRM

Procedure Name: LPR Main, LPR Main, LPRM Main

EZB0945E Could not send command (*SayCalRe*)

Explanation: The function which sends the commands was unsuccessful.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

User or Operator Response: Notify the system programmer

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPRM, LPQ

Procedure Name: LPR Main

EZB0946E Failed to read buffer (*SayCalRe*)

Explanation: The procedure that attempts to receive the data was unsuccessful.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPRM, LPQ

Procedure Name: LPR Main

EZB0947E TcpClose&colon (*SayCalRe*)

Explanation: The function that closes the connection was unsuccessful.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0948E Could not abort connection (*SayCalRe*)

Explanation: The application was unable to abort a connection. The reason is indicated by the *SayCalRe* portion of this message.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPR, LPQ, LPRM

Procedure Name: LPR Main

EZB0949E Failed to send from SendACK. Return Code = *rc* and Error Number = *errno*.

Explanation: The SendACK function, used by the server to relay acknowledgment, was not received by the host. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: Processing continues.

User or Operator Response: See *OS/390 IBM Communications Server: IP and SNA Codes* for information on return codes and error numbers.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendACK

EZB0950E Failed to send from SendCommand. Return Code = *rc* and Error Number = *errno*.

Explanation: The function that sends the acknowledgment was unsuccessful. The command was not sent. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See *OS/390 IBM Communications Server: IP and SNA Codes* for information on return codes and error numbers.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendCommand

EZB0951E Did not receive ACK for receive control file command

Explanation: SendCommand did not receive an acknowledgment code from the receiving host, and the control file command was not received.

System Action: LPR halts.

User or Operator Response: Retry the command making sure the host is ready and the connection is open.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0952E Failed to send control line block. Return Code = *rc* and Error Number = *errno*.

Explanation: The function that sends the data was unsuccessful. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See *OS/390 IBM Communications Server: IP and SNA Codes* for information on return codes and error numbers.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0953E Did not receive ACK for control file

Explanation: The function TcpFReceive returned an FRECEIVEerror, indicating that the receive request was rejected. The control file was not received.

System Action: LPR halts.

User or Operator Response: Make sure the receiving host is ready and retry the request.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0954E File could not be opened.

Explanation: LPR was unable to open a data set.

System Action: Processing continues.

User or Operator Response: Verify that the data set is in storage accessible to LPR.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0955E File contains no valid print lines.

Explanation: The data set is empty.

System Action: Processing continues.

User or Operator Response: Make sure the data set you wish to print contains at least 1 character.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0956E Failed to send data block (*SayCalRe*)

Explanation: The attempt to send the information was unsuccessful.

Note: For more information about this message, see the message **EZA***nnnn*, where *nnnn* is the 4-digit number in parentheses at the end of this message.

System Action: Processing continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Respond as indicated by the message EZA*nnnn* where *nnnn* is the 4-digit number in parentheses at the end of this message.

Source Data Set: LPR

Procedure Name: SendLineFlush

EZB0957E Did not receive ACK for receive data file command. Code = *code*.

Explanation: The host did not receive an acknowledgment from the LPR receive data command.

System Action: LPR halts.

User or Operator Response: See *OS/390 IBM Communications Server: IP and SNA Codes* for information on return codes and error numbers. The current port number and the user IP address are displayed with this message.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB0958E File could not be opened.

Explanation: The specified data set could not be opened.

System Action: LPR halts.

User or Operator Response: Verify that you have access to the data set and try to open it again.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB0959E Did not receive ACK for data file

Explanation: The remote host did not acknowledge that it received a data set.

System Action: If no acknowledgement is received, the data set is retransmitted. Processing continues.

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

System Programmer Response: Check why LPD did not send acknowledgement. Starting LPD (server) with "DEBUG" on, will help.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB0960E Did not receive ACK after close

Explanation: The remote host did not send a message acknowledging that it has closed. The connection remains open.

System Action: Processing continues.

User or Operator Response: Try to close the connection again.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DrainConnection

EZB0961I Control file name is *dataset name*

Explanation: The name of the control data set is displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: MakeFileNames

EZB0962I Data file name is *dataset name*

Explanation: The name of the current data set is displayed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: MakeFileNames

EZB0965E Use the form: *command name* **DataSetName.**

Explanation: This message displays the correct format for the indicated command. For more information about the commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: The command is not processed.

User or Operator Response: Reenter the command using the form given in the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: ProcessOperands

EZB0967E Data Set name "*name***" invalid.**

Explanation: The specified data set name is incorrect.

System Action: The command is not processed.

User or Operator Response: Make sure that the data set name uses the correct format and syntax and that the data set is in storage accessible to the host.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPD, LPR

Procedure Name: ProcessOperands

EZB0968E Data Set "*name***" not found or inaccessible, return code =** *rc***.**

Explanation: This message may issue one of the following return codes:

Return code	**Description**
4	Data set name not found
12	Data set name is missing
24	Data set is an unprintable VSAM file

If another return code is issued, this indicates that the data set attributes may make the data set inaccessible.

System Action: LPR halts.

User or Operator Response: Reenter the LPR command using the correct data set name. If the data set name is correct, check the data set attributes.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: ProcessOperands

EZB0969E Data Set "*name***" does not contain member "***member***".**

Explanation: The specified data set name does not contain the specified member name.

System Action: The LPR command is not processed.

User or Operator Response: Reenter the command using the correct data set name and the correct member name.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOperands

EZB0970E Data Set "*name*" invalid organization.

Explanation: The function FindDSName exited abnormally because the file's organization was incorrect.

System Action: Processing continues.

User or Operator Response: Recheck the organization and reenter the file. Notify the system programmer if the problem persists.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: ProcessOperands

EZB0971E Use a matching quotation mark at the end of the string.

Explanation: There is a missing quotation mark at the end of the string.

System Action: The LPR command is not processed.

User or Operator Response: Make sure there are appropriate quotation marks and reenter the string.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: LPRToken

EZB0973E Use either BINARY or NOBINARY but not both.

Explanation: Either BINARY or NOBINARY can be chosen. Data cannot be sent as BINARY and NOBINARY.

System Action: The LPR command is not processed.

User or Operator Response: Reenter the LPR command specifying either the BINARY or NOBINARY options.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0974E Use a token after the CLASS option.

Explanation: After the CLASS option, you must enter a name of a host. If you do not want to name another host name, the CLASS option is not required. For more information on the LPR command and its options, refer to the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: LPR command is not processed.

User or Operator Response: Reenter the LPR command and the CLASS option using a name of a host.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0975E Use a count after the COPIES option.

Explanation: This message indicates that there is no number following the COPIES option. The COPIES option specifies how many copies are to be printed. If this option is not used, one copy will be printed.

System Action: LPR command is not processed.

User or Operator Response: Reenter the LPR command with the COPIES option and fill in a number, specifying the number of copies you want printed. If only one copy is needed, the COPIES option is not required.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0976E Use a letter after the FILTER option.

Explanation: No filter was specified after the filter option in the LPR command. The filter option specifies the type of processing to be done with the data.

System Action: The LPR command is not processed.

User or Operator Response: Reenter the LPR command using the filter option with a valid filter. See *OS/390 IBM Communications Server: IP User's Guide* for more information on the LPR command.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0977E Use a number after the INDENT option.

Explanation: A numeric argument was not specified for the INDENT option of the LPR command.

System Action: LPR command is not processed.

User or Operator Response: Reenter the LPR command, specifying a numeric argument for the INDENT option. For more information about the LPR command and the INDENT option, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0978E Use a token after the JOB option.

Explanation: You must specify a name when using the JOB option. The name is the printing job's description to the remote system.

System Action: The LPR command is not processed.

User or Operator Response: Reenter the LPR command with a name following the JOB option. For more information on the JOB option of the LPR command see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0979E Use a number after the LINECOUNT option.

Explanation: You must enter a number following the LINECOUNT option to specify the number of lines to be printed before a new heading is printed.

System Action: The LPR command is not processed.

User or Operator Response: Reenter the LPR command with the LINECOUNT option using a valid number. For more information see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0980E Use an identification after NAME option.

Explanation: You did not specify a name after the NAME option in the LPR command. (The NAME option specifies the job information to be provided by the remote system in response to a query).

System Action: The LPR command is not processed.

User or Operator Response: Resubmit the LPR command supplying a name after the NAME option. For more information about the LPR command and the NAME option, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0981E Use a title after the TITLE option.

Explanation: You did not specify the title after the TITLE option in the LPR command.

System Action: The LPR command is not processed.

User or Operator Response: Resubmit the LPR command supplying a name after the TITLE option. For more information about the LPR command and the TITLE option, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0982E Use a number after the TOPMARGIN option.

Explanation: You did not specify a number after the TOPMARGIN option in the LPR command. (This number specifies the number of lines designated for the top margin).

System Action: The LPR command is not processed.

User or Operator Response: Resubmit the LPR command supplying the number of lines to be designated for the top margin. For more information about the LPR command and the TITLE option, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0983E Use a title after the WIDTH option.

Explanation: You did not specify the line width after the WIDTH option in the LPR command.

System Action: The LPR command is not processed.

User or Operator Response: Resubmit the LPR command supplying the line width of the data set. For more information about the LPR command and the WIDTH option see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0984E Use either the NOPOSTSCRIPT or POSTSCRIPT option but not both.

Explanation: You selected both the POSTSCRIPT and the NOPOSTSCRIPT options in the LPR command. You can select only one of these options.

System Action: The LPR command is not processed.

User or Operator Response: Reissue the command with the correct option. For more information about the LPR command and the POSTSCRIPT or NOPOSTSCRIPT options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0985E Use either the POSTSCRIPT or LANDSCAPE option but not both.

Explanation: You selected both the POSTSCRIPT and LANDSCAPE options in the LPR command. You can select only one of these options.

System Action: The LPR command is not processed.

User or Operator Response: Reissue the command with the correct option. For more information about the LPR command and the POSTSCRIPT or NOPOSTSCRIPT options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0986E Use either the BINARY OR LANDSCAPE option but not both.

Explanation: You selected both the BINARY and LANDSCAPE options in the LPR command. You may select only one of these options. If you select the LANDSCAPE option, the only valid filter option that you can specify explicitly is the ″o″ filter.

System Action: The LPR command is not processed.

User or Operator Response: Reissue the command using the correct option. For more information about the LPR command and the BINARY or LANDSCAPE options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0987E Use either the FILTER or LANDSCAPE option but not both.

Explanation: You selected both the FILTER and LANDSCAPE options in the LPR command. You may select only one of these options. If you select the LANDSCAPE option, the only valid filter option that you can specify explicitly is the ″o″ filter.

System Action: The LPR command is not processed.

User or Operator Response: Reissue the command with the correct option. For more information about the LPR command and the FILTER or LANDSCAPE options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0988I PostScript program is *number* bytes.

Explanation: This message specifies the number of bytes in the PostScript program.

System Action: The LPR command is not processed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB0989I Use these options.

EZB0990I *additional options*

EZB0991I *additional options* **or** *last options*

Explanation: TCPIP issues this message when one or more incorrect options is specified in the LPR command. It provides a list of valid options.

System Action: The LPR command is not processed.

User or Operator Response: Reissue the LPR command using valid options.

System Programmer Response: None.

Source Data Set: LPR, LPRSET

Procedure Name: ProcessOptions

EZB0992I File contains PostScript.

Explanation: TCPIP issues this message when tracing is on and PostScript characters are detected in the data set.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0993E Use the program which produced the PostScript to change to landscape display.

Explanation: The LANDSCAPE option is specified in the LPR command and the file to be printed is a PostScript data set.

System Action: The LPR command is not processed.

User or Operator Response: Redefine the data set for landscape display using the original application program that created the POSTSCRIPT file. Then reissue the LPR command without the LANDSCAPE option. See also EZB1010E.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0994E Use the NOPOSTSCRIPT option to prevent special processing of PostScript files.

Explanation: The LPR command contains both POSTSCRIPT and CC options.

System Action: The LPR command is not processed.

User or Operator Response: Reissue the LPR command using the NOPOSTSCRIPT option in place of the

POSTSCRIPT option. For more information about the LPR command and the POSTSCRIPT and NOPOSTSCRIPT options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0995W Ignoring "*buffer*"

Explanation: TCPIP issues this message when an incorrect carriage control character is detected. The lines containing those incorrect characters are ignored.

System Action: Processing continues.

User or Operator Response: Correct the carriage control characters in the data set.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0996W *number* **lines with invalid carriage control characters deleted.**

Explanation: TCPIP issues this message when incorrect carriage control characters are detected in the data set being processed. The lines containing the incorrect carriage control characters are deleted.

System Action: Processing continues.

User or Operator Response: Correct the carriage control characters in the data set and resubmit the job.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB0997I Byte size check starts at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The date and time are recorded before performing a byte size check of the data set.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB0998I Byte size check ends at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The date and time are recorded after performing a byte size check of the data set.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB0999I Send command starts at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. Arguments are passed prior to the actual data transfer. The date and time are recorded before those arguments are passed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1000I Send command ends at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. After arguments are passed and acknowledgments are received, the date and time are recorded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1001I Send data starts at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. Before the data set is sent to the remote printer, the date and time are recorded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1002I Send data ends at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. When the data set is received by the remote printer, the date and time are recorded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1003I Send ACK starts at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. TCPIP initiates an acknowledgment process. The date and time are recorded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1004I Send ACK ends at *date time*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. When the requested acknowledgments are received, the acknowledgment process is complete; the date and time are recorded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1005I Draining the connection.

Explanation: This message occurs while the data is sent from the host to the remote printer and tracing is on. Before the connection between the host and the remote printer closes, the information is taken in and processed by the printer.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DrainConnection

EZB1006E Host *host* **did not accept printer name** *printer*

Explanation: The indicated host did not recognize the indicated printer.

System Action: The LPR command is not processed.

User or Operator Response: Resubmit the job using the correct host printer name.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DrainConnection

EZB1007I Connection still receiving - aborting it.

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The remote host continues to receive information. When all the information is received, the connection is aborted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DrainConnection

EZB1008I Connection aborted.

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. TCPIP ended the connection with the remote host.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: DrainConnection

EZB1009I Data file sent.

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The data set has been successfully sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1010E Use the program which produced the PostScript to control pagination.

Explanation: This message follows EZB0993E and is issued when the LANDSCAPE option is specified in the LPR command and the file to be printed is a PostScript data set.

System Action: The LPR command is not processed.

User or Operator Response: Redefine the data set to control pagination using the original application program that created the POSTSCRIPT file. Then reissue the LPR command without the LANDSCAPE option.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB1011I Queuing control line "*number*"

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The commands are retrieved in the order in which they were submitted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: QueueControlLine

EZB1012I Receiving ACK

Explanation: TCPIP issues this message when tracing is on and a data set is sent to a remote printer using the LPR command. TCPIP is receiving acknowledgments from the remote host.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ReceiveByte

EZB1013I ReceiveACK: *word* **for byte value** *rc*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. TCPIP is receiving acknowledgments from the remote host.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ReceiveByte

EZB1014I Sending ACK

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The remote host is sending acknowledgments to TCPIP.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendACK

EZB1015I ACK successfully sent

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. The acknowledgment process has been successfully completed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendACK

EZB1016E Did not receive an ACK for command. Code=*rc*

Explanation: TCPIP issues this message when tracing is on and a data set is sent to the remote printer using the LPR command. TCPIP did not receive an acknowledgment from the remote host. The return code is displayed.

System Action: The LPR command is not processed.

User or Operator Response: Resubmit the LPR command.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendCommand

EZB1017I Control data sent

Explanation: A control data set was sent to the remote host. The control data set contains the printer setup information.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB1018I Control file sent

Explanation: A control data set was sent to the remote host. The control data set contains the information to be printed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendControlFile

EZB1019E Use the form: LPRSET nameofPrinter@printerhost.

Explanation: You specified an incorrect format for the LPRSET command.

System Action: The LPRSET command is not processed.

User or Operator Response: Reissue the LPRSET command with the correct syntax (nameofPrinter@printerhost). For more information about the LPRSET command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR, LPRSET

Procedure Name: ProcessOperands

EZB1020I Your LPR printer is currently set to *printer* **at** *host*

Explanation: This message is displayed when the `LPRSET (QUERY` command is issued. The LPR printer name and host are identified.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPRSET

Procedure Name: ProcessQueryOption

EZB1021I Append of host name to LASTING.GLOBALV status *rc*

Explanation: This message is displayed when the LPRSET command is issued while tracing is on. The remote host name is added to the LASTING.GLOBALV data set. This message is followed by EZB1022I and EZB1023I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPRSET

Procedure Name: ProcessArguments

EZB1022I Append of printer name to LASTING.GLOBALV status *rc*

Explanation: This message is displayed when the LPRSET command is issued while tracing is on. The printer name is added to the LASTING.GLOBALV data set. This message appears with EZB1021I and EZB1023I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPRSET

Procedure Name: ProcessArguments

EZB1023I Printer set to *printer* **at** *host*

Explanation: This message is displayed when the LPRSET command is issued while tracing is on. The LPR printer name and host are identified. This message is preceded by EZB1021I and EZB1022I.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPRSET

Procedure Name: ProcessArguments

EZB1024E System error *error* **setting printer.**

Explanation: This message occurs due to a failure while attempting to set the LPR printer. For more information about the LPRSET command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: The LPRSET command is not processed.

User or Operator Response: Check the remote printer and resubmit the job.

System Programmer Response: None.

Source Data Set: LPRSET

Procedure Name: ProcessArguments

EZB1025E Cannot use BINARY with a DBCS Translation Mode.

Explanation: The BINARY parameter may not be specified with any other of the following LPR command line parameters:
- JIS78KJ
- JIS83KJ
- SJISKANJI
- EUCKANJI
- IBMKANJI
- HANGEUL
- KSC5601
- TCHINESE

System Action: The LPR command is not processed.

User or Operator Response: Consult your system programmer.

System Programmer Response: Specify either the BINARY parameter or a DBCS translation mode parameter. See the *OS/390 IBM Communications Server: IP User's Guide* for more information about using DBCS conversion in LPR.

Source Data Set: LPR

Procedure Name: ProcessDbcsOption

EZB1026E Cannot use multiple DBCS Translation Modes.

Explanation: More than one of the following LPR command line parameters was specified:
- JIS78KJ
- JIS83KJ
- SJISKANJI
- EUCKANJI
- IBMKANJI
- HANGEUL
- KSC5601
- TCHINESE

System Action: The LPR command is not processed.

User or Operator Response: Consult your system programmer.

System Programmer Response: Specify only one of the above DBCS translation mode parameters. See the *OS/390 IBM Communications Server: IP User's Guide* for more information about using DBCS conversion in LPR.

Source Data Set: LPR

Procedure Name: ProcessDbcsOption

EZB1027E Unable to Load *DBCS_translate_table_dataset*

Explanation: LPR was attempting to load a DBCS translation table corresponding to the DBCS conversion mode parameter specified on the LPR command line. All data sets in the search order hierarchy for the required translate table dataset, either do not exist, or do not contain data in the required format for DBCS binary translate tables.

System Action: The LPR command is not processed.

User or Operator Response: Consult your system programmer.

System Programmer Response: Configure a valid DBCS binary translate table data set in the search order hierarchy for the required DBCS translation table. Refer to the *OS/390 IBM Communications Server: IP User's Guide* for more information about the loading and customizing of DBCS translation tables.

Source Data Set: LPR

Procedure Name: LoadDbcsTables

EZB1043E Failed to Send PostScript_id. Return code = *rc***. Error Number=** *errno***.**

Explanation: This message is issued as a result of the LPD server failing to send the PostScript id contained within the PostScript data file.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: Notify the system programmer.

System Programmer Response: See the return code and error number issued in this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* to determine and correct the error.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1044E Specify a user name after USER option.

Explanation: The LPR command was entered with no user name specified for the USER option.

System Action: LPR halts.

User or Operator Response: Reenter the LPR command, specifying a user name after the USER option. For more information, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB1045E Send ACK failed. Return Code = *rc*. Error Number = *errno*.

Explanation: The client machine failed while attempting to send acknowledgments to the server machine. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the return code and error number issued in this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* to determine and correct the error.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1046E Shutdown failed. Return Code = *rc*. Error Number = *errno*.

Explanation: While attempting to close the TCP connections on both the client and server sides, an error was detected. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the return code and error number issued in this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* to determine and correct the error.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1047E Send Control File failed. Return Code = *rc*. Error Number = *errno*.

Explanation: An error was detected while attempting to send the file format information. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the return code and error number issued in this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* to determine and correct the error.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1048E Send Data File failed. Return Code = *rc*. Error Number = *errno*.

Explanation: An error was detected while attempting to send the data file containing the file data. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the return code and error number issued in this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* to determine and correct the error.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1049E Send printer command did not receive ACK. ACK message = *message*.

Explanation: No acknowledgment was received from the server while attempting to send a printer command. An acknowledgment message was passed. The current port number and the user IP address are displayed with this message.

System Action: LPR halts.

User or Operator Response: Use the ACK message to determine and correct the problem.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1050E Failed to Send Data. Return Code = *rc*. Error Number = *errno* port number = *port* remote ip address = *ipaddr*

Explanation: An error was detected while attempting to send data to the server machine. The current port number and the user's IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: Look up the *error number* issued in this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* for help in diagnosing the exact problem.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1051E Failed to Open connection to Port Number = *port*. return code = *rc* error number = *errno* port number = *port* remote ip address = *ipaddr*

Explanation: The attempt to open a connection at the stated port was unsuccessful. The requested port, return code, error number, current port, and the IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: Try to access the stated port number again. If the problem persists, notify the system programmer.

System Programmer Response: Look up the *errno* issued with this message in the *OS/390 IBM Communications Server: IP and SNA Codes*for help in diagnosing the exact problem.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1052E StartTcpip: Return Code = *rc* and Error Number = *errno*.

Explanation: An error was detected while attempting to initialize TCPIP.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes

- *OS/390 IBM Communications Server: IP and SNA Codes*

3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the error number and return code issued with this message and refer to the *OS/390 IBM Communications Server: IP and SNA Codes* for more information on error numbers and return codes.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1053E Could not set option. Return Code = *rc* **and Error Number =** *errno***.**

Explanation: While attempting to set the socket KeepAlive option, an error was detected.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the *OS/390 IBM Communications Server: IP and SNA Codes* for information on return codes and error numbers.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1054E TcpClose: Return Code = *rc* **and Error Number =** *errno***.**

Explanation: While attempting to close the TCP connection on both the client and server machines an error was detected. The current port number and the user IP address are displayed with this message.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: LPR halts.

User or Operator Response: See the *OS/390 IBM Communications Server: IP and SNA Codes* for more information on return codes and error numbers.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1055E Use a three-digit number after the JNUM option.

Explanation: The value specified for JNUM is not valid. It is either missing, greater than or less than three digits, or is not numeric. The JNUM value may be any three-digit number between 000 and 999. The print job is not processed.

System Action: LPR halts.

User or Operator Response: Correct the JNUM value specified on the LPR command.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB1056E Invalid Internet address *Foreignhost***.**

Explanation: You provided an address containing numbers greater than 255. This is an invalid Internet address for LPR.

System Action: LPR halts.

User or Operator Response: Check the Internet address to which you are sending your output.

System Programmer Response: Assist the user as necessary.

Source Data Set: LPR

Procedure Name: MainLoop

EZB1057I Loaded translation table from *data set name*

Explanation: The fully qualified data set name used by LPR for the translation tables is displayed. These tables are used to translate EBCDIC to ASCII and ASCII to EBCDIC. This message may be issued more than once if more than one table is loaded. The last table loaded of a particular type will be the one used. (This will be the one cited in the last occurrence of this message).

System Action: Processing continues.

User or Operator Response: If the data set name displayed is not the one desired, verify that the expected data set exists. The search order used by LPR to find the translation data set is described in the *OS/390 IBM Communications Server: IP Configuration Reference*. If the data set name displayed is empty (″ ″), then no translate data set was found in the search order. Create the desired translate table, or correct the TRANSLATE option specified on the LPR command.

System Programmer Response: Verify that the system translate tables are correctly installed under the correct high level qualifier (e.g. *hlq*.STANDARD.TCPXLBIN). For information on determining the high level qualifier used, refer to the *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: LPR

Procedure Name: ProcessOptions

EZB1098I Return Code = *rc***. Error Number =** *errno***.**

Explanation: See the user response. The text of this message is usually appended to the end of other messages.

About ERRNOs and ERRNOJRs:

1. If the code is 8 characters long, the first 4 characters are the product identifier and the last 4 characters are the actual code.
2. These codes can be found in either of the following:
   - *OS/390 UNIX System Services Messages and Codes* where ERRNOs are called return codes and ERRNOJRs are called reason codes
   - *OS/390 IBM Communications Server: IP and SNA Codes*
3. The codes might be in either hex or decimal.

System Action: Depending on the return code and error number in this message, LPR process may continue or may terminate.

User or Operator Response: See the *OS/390 IBM Communications Server: IP and SNA Codes* for more information on return codes and error numbers. If the return code = -1 and error number = 0, then negative acknowledgment was received from the server. Turn on the trace for the server to find out why the server rejected the LPR request.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1099I Port Number = *port*. **Remote IP Addr =** *target address*

Explanation: See the user response. The text of this message is usually appended to the end of other messages. This message displays the port number and the IP address currently in use.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LPR

Procedure Name: SendDataFile

EZB1100E Cannot load translate table *data_set*

Explanation: LPR was attempting to load an SBCS translation table corresponding to the TRANSLATETABLE or XLATETABLE parameter specified on the LPR command line. All data sets in the search order hierarchy for the required translate table data set, either do not exist, or do not contain data in the required format for SBCS binary translate tables.

System Action: LPR halts.

User or Operator Response: Consult your system programmer.

System Programmer Response: Configure a valid SBCS binary translate table data set in the search order hierarchy for the required SBCS translation table. See the *OS/390 IBM Communications Server: IP Configuration Reference* for more information about the loading and customizing of SBCS translation tables.

Source Data Set: LPR

Procedure Name: ProcessOptions

Chapter 3. EZB1200—EZB1230

X Window Messages

This section contains X Window messages.

EZB1200E Error parsing argument "*option*" (*specifier*) ; unknown kind

Explanation: XWindows has encountered an option or specifier that it does not recognize in a user-submitted command line. The command line is not processed.

System Action: XWindows continues.

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

System Programmer Response: None.

Source Data Set: PARSECMD

Procedure Name: _XReportParseError

EZB1201E Bad image *image_type*: *error* found in routine: *routine*

Explanation: Xwindows received an incorrect image type from the indicated routine. The image type and Xwindows error code are displayed in the message.

System Action: Xwindows halts.

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

System Programmer Response: Use the error code given in the message and *OS/390 IBM Communications Server: IP and SNA Codes* to determine the cause of the error, and correct the indicated routine as necessary.

Source Data Set: XIMUTIL

Procedure Name: _XReportBadImage

EZB1202E Xlib: warning, client built for newer rev (*rev_number*) than server (*rev_number*)!

Explanation: The Xwindows client code was built for a newer version of Xwindows than the Xwindows server code. This can result in compatibility errors.

System Action: Xwindows halts.

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

System Programmer Response: Correct either the client code or the server code to eliminate the revision disparity and rebuild the code.

Source Data Set: XOPENDIS

Procedure Name: _XRead

EZB1203E Xlib: warning, client is protocol rev *revision*, server is rev *revision*!

Explanation: The Xwindows server is using a different revision of the Xwindows code than the client. This can cause compatibility problems.

System Action: Xwindows halts.

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

System Programmer Response: Correct the disparity in revision levels between the client and the server and restart Xwindows.

Source Data Set: XOPENDIS

Procedure Name: _XRead

EZB1204E Xlib: connection to *server* refused by server

Explanation: This message indicates that no authorization exists between the Xwindow server and the client program. The error causes the server to refuse the connection. The connection is not established.

System Action: No Xwindows session can be opened for this client.

User or Operator Response: Issue an Xhost command from the Xserver side.

System Programmer Response: None.

Source Data Set: XOPENDIS

Procedure Name: Main

EZB1205E Xlib: sequence lost (0x*number_received* > 0x*number_expected*) in reply type 0x*reply_type*!

Explanation: The Xwindows server received a reply packet with a sequence number greater than the sequence number expected, indicating that a reply packet has been lost.

System Action: Xwindows halts processing.

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

System Programmer Response: Check the indicated Xclient to determine why the packet was lost and respond as indicated.

Source Data Set: XLIBINT

Procedure Name: _XSetLastRequestRead

EZB1206E Xlib: unhandled wire event! event number = *number*, display = *display_number*

Explanation: Xwindows encountered an unknown event during conversion between the host format and the wire format.

System Action: Xwindows halts.

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

System Programmer Response: Check the XEvent structure to make sure it is compatible with the current host. For more information, refer to the Xwindows documentation.

Source Data Set: XLIBINT

Procedure Name: _XUnknownWireEvent

EZB1207E Xlib: unhandled native event! event number = *number*, display = *display_number*

Explanation: Xwindows encountered an unknown event while reformatting a wire event to the host structure.

System Action: Xwindows halts.

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

System Programmer Response: Check the XEvent structure to make sure that it is compatible with the current host. For more information refer to the Xwindows documentation.

Source Data Set: XLIBINT

Procedure Name: _XUnknownNativeEvent

EZB1208E XIO: fatal IO error *error_description* on X server *address_of_server*

EZB1209E after *number* **requests (***number* **known processed) with** *number* **events remaining.** *err_no* **(***error***) on X server "***server***"**

Explanation: The Xwindows server encountered an I/O error during processing. The type of error is indicated in the message. The error description portion of this message indicates the cause of the error.

System Action: Xwindows halts.

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

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

Source Data Set: XLIBINT

Procedure Name: _XDefaultIOError

EZB1210I The connection was probably broken by a server shutdown or KillClient.

Explanation: If this message follows messages EZB1208E and EZB1209E, it indicates that the error displayed in those messages was probably caused by the shutdown of the Xwindows server, or by the procedure KillClient, which closes a connection to an Xwindows client.

System Action: Xwindows halts.

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

System Programmer Response: Check the Xwindows server to determine why it shut down or closed the client connection.

Source Data Set: XLIBINT

Procedure Name: _XDefaultIOError

EZB1211I Request failed due to IUCV error.

Explanation: This message follows messages EZB1208E and EZB1209E, and indicates that the error displayed in those messages was caused by a failure in the inter-user communication vehicle (IUCV) Xwindows uses IUCV to transmit requests and receive replies.

System Action: Xwindows halts.

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

System Programmer Response: See additional messages to determine the cause of the IUCV error and respond as indicated. client connection.

Source Data Set: XLIBINT

Procedure Name: _XDefaultIOError

EZB1212E Xlib: extension "*extension***"** *reason* **on display "***display***".**

Explanation: Xwindows encountered an incorrect event handler when the function XSetExtensionErrorHandler or XMissingExtension is called.

System Action: Xwindows halts.

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

System Programmer Response: Make sure that the event handler is set correctly before calling the associated extension. For more information refer to the Xwindows documentation.

Source Data Set: EXTUTIL

Procedure Name: XExtDisplayInfo

EZB1220I **name:** *argument*, **value: 0x***hex_value*

Explanation: This message displays the name and hexadecimal value of arguments for Xwindows processing.

System Action: Xwindows continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ARGLIST

Procedure Name: PrintArgList

EZB1221E *error_prefix***Error:** *reason*

Explanation: Xwindows has encountered an error. The reason for the error is described in the *reason* portion of this message.

System Action: Xwindows halts.

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

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

Source Data Set: ERROR

Procedure Name: _XtDefaultError

EZB1222E *warning_prefix* **Warning:** *reason*

Explanation: Xwindows has encountered an error. The reason for the error is described in the *reason* portion of this message.

System Action: Xwindows halts.

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

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

Source Data Set: ERROR

Procedure Name: _XtDefaultWarning

EZB1230I *header* **w:** *children*, **m:** *request_mode*, **x:** *x_coordinate*, **y:** *y_coordinate*, **w:** *width*, **h:** *height*, **b:** *border_width*

Explanation: This message displays information about the geometry of the current Xwindows.

System Action: Xwindows continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: GEOUTILS

Procedure Name: PrintBox

Chapter 4. EZB2000—EZB2498

XNX25 Messages

This section contains XNX25 messages.

EZB2000I *string string hexdata*

Explanation: The contents of a packet or control block are displayed.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: SNAPIA5, SNAPAREA

EZB2010I *program* **{MVS} update level** *level*

Explanation: The version of X25IPI currently running has the indicated program name (usually X25IPI) and the indicated update level.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the update level when reporting errors.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2011I **Command:** *command*

Explanation: This message is displayed when tracing is on. The X25 server is processing the command passed to it from the console input program.

System Action: The command is processed and execution continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: COMMAND

EZB2012E **Unrecognized command:** *command*

Explanation: The X25 server tried to process an unrecognized command that was passed to it from the console input program.

System Action: The X25 server continues processing.

User or Operator Response: Correct the command and resubmit. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information about TCPIP X25 commands.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: COMMAND

EZB2013T Unrecognized CIB verb= *cibverb*

Explanation: The X25IPI server received an unexpected command verb in a command input buffer from the command queue (QEDIT). The valid commands are STOP, START, and MODIFY.

System Action: X25IPI treats the command as STOP and exits.

User or Operator Response: Restart TCPIPX25.

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

Source Data Set: XNX25IPI

Procedure Name: COMMAND

EZB2020I MCH *lu* **state** *state session_state*

Explanation: X25IPI status for the indicated MCH is displayed. The following are the *Link_state* fields:

Field	Description
X'00'	Restart needed
X'10'	Ready
X'20'	Restart request sent
X'30'	Restart indication received

The following are the VTAM®*session_state* fields:

Field	Description
X'20'	Enabled for LOGAPPL logon
X'30'	logon pending
X'40'	Opening
X'50'	Open (ready)
X'60'	Closing
X'70'	Unsuccessful

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the status in problem determination.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2021I VC *vc* **LU** *lu* **DTE** *address* **state** *call_state session_state*

Explanation: X25IPI status for the indicated virtual circuit is displayed. The following are the *call_state* fields:

Field	Description
X'00'	Waiting for restart on MCH
X'10'	Available for connections
X'20'	Call request sent
X'30'	Call request received
X'40'	Ready for data transfer
X'41'	Reset request sent
X'42'	Reset request received
X'43'	Interrupt request sent
X'44'	Interrupt request received
X'50'	Call collision
X'60'	Clear request sent
X'70'	Clear request received

The following are the VTAM*session_state* fields:

Field	Description
X'00'	Disabled until restart on MCH

X'10' Available for connections
X'20' Enabled for NPSI LU logon
X'30' Logon pending
X'40' Opening
X'50' Open (ready)
X'60' Closing
X'70' Failed

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the status in problem determination.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2022R IP *AS_asname* **state** *connection status*

Explanation: X25IPI status for the DLC path to the TCPIP address space is displayed. The following are the state codes:
X'80' DLC connection complete
X'40' DLC connection pending
X'00' no path

System Action: The X25 server continues processing.

User or Operator Response: Use the status in problem determination.

System Programmer Response: Use the status in problem determination.

Source Data Set: XNX25IPI

Procedure Name: CLIST

EZB2030I MCH *lu* **RC:** *name* **IP:** *name* **SN:** *name* **TX:** *name*

Explanation: In response to an EVENTS command, the internal subroutine names handling events associated with this MCH are displayed.

Subroutine	Description
IP	VTAM input pending
RC	VTAM request complete
SN	VTAM session notification
TX	Timer expired

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the data for problem determination.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2031I VC *vc* **RC:** *name* **IP:** *name* **SN:** *name* **TX:** *name*

Explanation: In response to an EVENTS command, the internal subroutine names handling events associated with this virtual circuit is displayed.

Subroutine	Description
RC	VTAM request complete
IP	VTAM input pending
SN	VTAM session notification
TX	Timer expired

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the data for problem determination.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2032I MCH *lu*

Explanation: The following X25IPI014R VC messages apply to virtual circuits on this MCH.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2033I VC *vc DTE_address* **S** *send_count* **R** *receive_count* **D** *drop_count* **Q** *queue_size*

Explanation: The traffic counts for this virtual circuit are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2034I IP *AS_userid* **S** *send_count* **R** *receive_count* **D** *drop_count* **Q** *queue_size*

Explanation: The traffic counts on the IUCV connection to the TCPIP address space are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2040E MCH *lu* **unknown**

Explanation: An attempt was made to send a call request for an unknown MCH. The request is ignored. The CERTCALL command is refused.

System Action: TCPIP continues.

User or Operator Response: Determine the correct MCH name, and reissue the CERTCALL command.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2041I MCH *lu* restarting

Explanation: X25IPI is restarting this MCH in response to a RESTART command. A VTAM session is initiated for the NPSI MCH LU.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2042E MCH *lu* unavailable

Explanation: X25IPI could not restart this MCH because it is unavailable for logon. The MCH session is left available for initiation by LOGAPPL.

System Action: TCPIP continues.

User or Operator Response: Activate the MCH LU through VTAM.

System Programmer Response: Determine the reason why VTAM refused session initiation.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2043E MCH *lu* already started

Explanation: X25IPI could not restart this MCH because it is already active. The request is ignored.

System Action: TCPIP continues.

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

System Programmer Response: Determine the correct MCH name and reissue the CERTCALL command.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2050I *object* **at** *address additional*

Explanation: This message reports the locations of internal storage as requested by debug flag 4. The *object* can be ″Global storage″, ″MCH *lu* SDA″, or ″VC *vc* SDA″. The *additional* variable, if present, reports the ″stack at *address* limit *address*″.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: Use the storage addresses for problem determination.

Source Data Set: XNX25IPI

Procedure Name: STRTLINK

EZB2051I MCH *lu* SDA at *address*

Explanation: The contents of the session data area (SDA) for this connection are dumped.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Use the data for problem determination.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2080I **Dispatch** *handler name* **(***event handler address***) SDA** *session data area address* **ECB** *event control block*

Explanation: This message is displayed when tracing is on. The X25 server scans for posted event control blocks and calls event handlers.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: Use the dispatch trace in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MAINLOOP

EZB2081I **Dispatch** *handler name* **(***event handler address***) SDA** *session data area address*

Explanation: This message is displayed when tracing is on. The X25 server scans for posted event control blocks and calls an MCH event handler.

System Action: The event handler is called.

User or Operator Response: None.

System Programmer Response: Use the dispatch trace in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MCHEVENT

EZB2082I **Main wait:** *number of ECBs* **ECBs**

Explanation: This debug message displays the number of event control blocks (ECB) in the main wait list.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: Use the event count in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MAINLOOP

EZB2083I **Call** *caller's name* **(***caller's address***) from** *previous caller* **in** *previous caller's name* **(***previous caller's address***)**

Explanation: This debug message displays the name of the current subroutine and the calling subroutine.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: Use the trace in problem determination.

Source Data Set: XNX25IPI

Procedure Name: FOLLOW

EZB2084E **Posted ECB at** *address* **had no handler**

Explanation: The X25IPI event control block at this address did not specify an event handler routine.

System Action: TCPIP continues.

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

System Programmer Response: If the error recurs, obtain a dump of the X25IPI address space to make sure that the correct event handler is loaded.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2085E ECB address being added twice to wait list

Explanation: The X25IPI routine that adds elements to the ECB wait list detected an attempt to add an ECB twice.

System Action: TCPIP continues.

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

System Programmer Response: Determine the external event that caused this condition. Recreate it with an X25 internal trace running. Obtain a dump of the X25 region and submit the dump to the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None

EZB2090I Terminating

Explanation: X25IPI is terminating its execution and shutting down.

System Action: The VTAM ACB is closed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2091I HALT notice accepted type *type*

Explanation: X25IPI received a VTAM HALT notification or a HALT console command and is shutting down execution.

System Action: VTAM LU sessions are closed, and the IUCV connection to the TCPIP address space is severed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2092T Stack overflow at *address*

Explanation: X25IPI encountered a stack overflow at the indicated address. ABEND message X'091' is displayed.

System Action: X25IPI abends.

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

System Programmer Response: Determine the cause of overflow, and submit the ABEND dump to the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2093T Buffer released twice at *address* **in** *routine* **(***address***)**

Explanation: X25IPI encountered a consistency error in its buffer allocation pool. A buffer was released twice. This indicates a programming error. Return code X'92'. is displayed.

System Action: X25IPI abends.

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

System Programmer Response: Determine the cause of the consistency error, and submit the ABEND dump to the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2094S NPSI SEND completion, pending packet = *packet number*

Explanation: A program flag indicating a deferred control packet to be sent has an unacceptable value. The pending condition is reset.

System Action: TCPIP continues.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2095T WTO text overflow

Explanation: XNX25IPI encountered an overflow in the Write To Operator (WTO) area, which is used to display informational and error messages. ABEND message X'099' is displayed.

System Action: The program abends.

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

System Programmer Response: Determine the cause of the overflow, and submit the ABEND dump to the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2099I Ended

Explanation: X25IPI has finished processing.

System Action: The X25IPI program exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2100I *configuration dataset record*

Explanation: This debug message displays a record that was read from the configuration data set.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: X25IPI1

EZB2101E Unable to open configuration file, DDNAME=X25IPI

Explanation: X25IPI encountered an error opening its X25IPI configuration data set.

System Action: X25IPI does not start.

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

System Programmer Response: Correct the DD for the X25IPI configuration data set.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2102E Unrecognized configuration entry: *keyword*

Explanation: X25IPI encountered an unrecognized entry in its X25IPI configuration data set. This configuration record is skipped; further configuration errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the entry in the X25IPI configuration data set.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2103E Missing configuration entry: *keyword*

Explanation: X25IPI could not find the indicated keyword in its configuration data set. You should have at least one entry defining a link in the file.

System Action: X25IPI does not start.

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

System Programmer Response: Add a Link definition to the X25IPI configuration data set for each indicated keyword to be used for TCPIP traffic.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2104E Buffer storage not available (*bytes* **bytes required)**

Explanation: X25IPI could not allocate sufficient storage during initialization.

System Action: X25IPI does not start.

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

System Programmer Response: Increase the X25IPI region or address space storage size to the indicated number of bytes.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2105I Unable to open TCPIP DATA file, DDNAME=X25IPITD; default TCPIP userid used

Explanation: X25IPI encountered an error opening the X25IPI configuration data set. X25IPI uses the default TCPIP user ID.

System Action: TCPIP continues.

User or Operator Response: If TCP connections cannot be made, notify the system programmer.

System Programmer Response: If the user ID of the TCPIP server is not TCPIP, define a DD for X25IPITD referencing the X25IPI configuration data set.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2106E DLC Init function X25IPI failed R15=*value*

Explanation: X25IPI encountered an error issuing a DLC Init call.

System Action: X25IPI does not start.

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

System Programmer Response: Check that the X25IPI procedure is in the PROCLIB.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2107T Programming error: not enough buffers allocated

Explanation: The calculation of the number of buffers to allocate was incorrect.

System Action: X25IPI does not start.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2110E Unrecognized trace level: *level*

Explanation: The trace level, specified in either the X25IPI configuration data set or the TRACE command, is not correct. The trace level is set to “off”.

System Action: TCPIP continues.

User or Operator Response: Reissue the TRACE command with valid trace level.

System Programmer Response: Correct the Trace entry in the X25IPI configuration data set.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2111I VTAM ACB *IPI_APPN* **opened successfully**

Explanation: The virtual transmission access method (VTAM) address space successfully opened an activity control block (ACB) for the indicated application.

System Action: The application begins processing. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: xnx25ipi

Procedure Name: main

EZB2112E Repeated VTAM record ignored: *record*

Explanation: X25IPI encountered more than one VTAM entry in the X25IPI configuration data set. The repeated VTAM record is skipped.

System Action: Initialization continues.

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

System Programmer Response: Remove the repeated VTAM record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2113E VTAM application name missing

Explanation: The VTAM application ID is missing on the VTAM entry in the X25IPI configuration data set.

System Action: X25IPI does not start.

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

System Programmer Response: Specify the VTAM application ID and password on the VTAM entry.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2114E VTAM application password missing

Explanation: The VTAM application password is missing on the VTAM entry in the X25IPI configuration data set.

System Action: X25IPI does not start.

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

System Programmer Response: Specify the VTAM application password on the VTAM entry.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2115E VTAM ACB *application* **open failed**

Explanation: X25IPI encountered an error opening the VTAM ACB. This message is preceded by a EZB2401E or EZB2402E message reporting the VTAM error code.

System Action: X25IPI does not start.

User or Operator Response: Activate the X25IPI application in VTAM.

System Programmer Response: Use the VTAM error code from the message EZB2401E or EZB2402E to determine why the VTAM ACB OPEN macro was unsuccessful.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2121E DDN and non-DDN links cannot be mixed

Explanation: X25IPI encountered Link statements in its X25IPI configuration data set specifying both DDN and non-DDN links. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Separate DDN and non-DDN links into 2 X25IPI virtual machines.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2122E Link LU name missing

Explanation: The NPSI MCH LU name is missing on the Link entry in the X25IPI configuration data set. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Complete the Link entry. Refer to *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2123E Link DTE address missing

Explanation: X25IPI encountered a Link statement without a DTE address specified. This field is required for all non-DDN links. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Complete the Link entry. Refer to *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2124E Link window size missing or not numeric

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that had a missing or nonnumeric default window size specification. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the window size. Refer to the *OS/390 IBM Communications Server: IP Configuration Reference* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2125E Link window size not in range 1..7 or 1..127

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set with an incorrect default window size specification. The window size should be in the range 1–7 or 1–127. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the window size.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2126E Link packet size missing or not numeric

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that had a missing or nonnumeric default packet size. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the packet size. Refer to *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2127E Link packet size unacceptable

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set with an incorrect default packet size specification. The size should be 32, 64, 128, 256, 512, 1024, 2048, or 4096. The Link entry is discarded; further errors can result.

System Action: The link entry is discarded. Processing continues.

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

System Programmer Response: Correct the packet size.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2128E Link logical channel count missing or not numeric

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that had a missing or nonnumeric logical channel count field. The Link entry is discarded; further errors can result.

System Action: The Link entry is discarded. Processing continues.

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

System Programmer Response: Correct the channel count. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2129E Link logical channel count not in range 1..1023

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set with an incorrect logical channel count field. The count should be in the range 1–1023. The Link entry is discarded; further errors can result.

System Action: The Link entry is discarded. Processing continues.

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

System Programmer Response: Correct the channel count.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2130E Link reserved channel count missing or not numeric

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that had a missing or nonnumeric reserved channel count. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the reserved channel count. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2131E Link reserved channel count not in range 1..*LCC*

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set with an incorrect reserved channel count. The count should be between 1 and the number of logical channels that have been defined. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the reserved channel count. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2132E Link data network code missing

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that omitted a data network identifier code (DNIC). DNICs should be specified for all networks (use DDN for DDN nets). The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Complete the Link entry. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2133E Data network identifier code not decimal

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that specified a nondecimal data network identifier code (DNIC). The DNIC should be decimal. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the DNIC number. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Link definition.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2134E Link DTE address not decimal

Explanation: X25IPI encountered a Link statement in its X25IPI configuration data set that contained a nondecimal DTE address. The Link entry is discarded; further errors can result.

System Action: Processing continues.

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

System Programmer Response: Correct the DTE address.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2135E Altlink record not preceded by a Link record

Explanation: The X25 server is processing an ALTLINK record that was not preceded by a LINK record in the configuration data set.

System Action: X25 ignores the ALTLINK configuration record.

User or Operator Response: Correct the configuration data set by either removing the ALTLINK statement or adding the correct LINK statement preceding the ALTLINK statement. For more information about the ALTLINK and LINK statements, see *OS/390 IBM Communications Server: IP Configuration Guide* .

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTALTL

EZB2140E Dest record not preceded by Link record

Explanation: X25IPI encountered a Dest record in its X25IPI configuration data set before any Link record. The Dest record should follow the corresponding Link record in the X25IPI configuration data set. The Dest entry is skipped.

System Action: Processing continues.

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

System Programmer Response: Move the Dest record after the corresponding Link record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2141E Dest IP address missing

Explanation: X25IPI encountered a Dest record in its X25IPI configuration data set that did not specify an IP address. The Dest entry is discarded.

System Action: Processing continues.

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

System Programmer Response: Complete the Dest entry. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Dest record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2142E Dest IP address must be decimal

Explanation: X25IPI encountered a Dest record in its X25IPI configuration data set specifying a nondecimal destination address. The destination address should be in dotted-decimal form. The Dest entry is discarded.

System Action: Processing continues.

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

System Programmer Response: Correct the IP address. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Dest record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2143E Dest DTE address missing

Explanation: X25IPI encountered a Dest record in its X25IPI configuration data set, which omitted the destination X.25 DTE address specification. The destination X.25 DTE address specification is required on non-DDN networks. The Dest entry is discarded.

System Action:

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

System Programmer Response: Complete the Dest record with the X.25 DTE address. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Dest record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2144E Dest DTE address must be decimal

Explanation: X25IPI encountered a Dest record in its X25IPI configuration data set that had a nondecimal destination X.25 DTE address specified. The destination X.25 DTE address should be decimal. The Dest entry is discarded.

System Action: Processing continues.

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

System Programmer Response: Correct the DTE address.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2145E Dest call user data must be hexadecimal

Explanation: X25IPI encountered a Dest record in its X25IPI configuration data set specifying nonhexadecimal call user data (CUD) protocol ID. The CUD should be hexadecimal. The Dest entry is discarded.

System Action: Processing continues.

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

System Programmer Response: Correct the call user data.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2146E Dest facilities data must be hexadecimal

Explanation: The X25 server is processing the facilities field on a DEST statement. The field contained a non-hexadecimal character.

System Action: X25 discards this record and continues.

User or Operator Response: Correct the configuration data set by correcting the DEST facilities field. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTDEST

EZB2150E Datagram size limit missing or not numeric

Explanation: X25IPI encountered a Buffers record in its X25IPI configuration data set that had a missing or nonnumeric buffer size specification. The Buffers record is ignored. The default values are applied.

System Action: Processing continues.

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

System Programmer Response: Complete the Buffers record with the datagram size limit. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Dest record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2151E Datagram size limit not in range 576...2048

Explanation: X25IPI encountered a Buffers record in its X25IPI configuration data set with an incorrect buffer size specification. The buffer size should be in the range 576–2048. The Buffers record is ignored. The default values are applied.

System Action: Processing continues.

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

System Programmer Response: Correct the datagram size limit.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2152E Extra buffer count missing or not numeric

Explanation: X25IPI encountered a Buffers record in its X25IPI configuration data set that had a missing or nonnumeric buffer count field. The extra buffer count is ignored.

System Action: The extra buffer count is ignored. Processing continues.

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

System Programmer Response: Correct the buffer count.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2153E VC send queue limit missing or not numeric

Explanation: X25IPI encountered a Buffers record in its X25IPI configuration data set that had a missing or nonnumeric send queue limit. The send queue limit defaults to 8.

System Action: Processing continues.

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

System Programmer Response: Correct the queue limit.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2155E Inactivity timeout missing or not numeric

Explanation: X25IPI encountered a Timers record in its X25IPI configuration data set that had a missing or nonnumeric inactivity time-out value. The Timers record is ignored. The default values are applied.

System Action: Processing continues.

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

System Programmer Response: Complete the Timers record with the inactivity time-out. See *OS/390 IBM Communications Server: IP Configuration Reference* for the format of the Dest record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2156E Minimum call timer missing or not numeric

Explanation: X25IPI encountered a Timer record in its X25IPI configuration data set that had a missing or nonnumeric minimum time field. This Timer record is ignored. The minimum call time defaults to 60 seconds.

System Action: Processing continues.

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

System Programmer Response: Correct the minimum call time.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2160E Options record not preceded by Link record

Explanation: X25IPI encountered an Options record in its X25IPI configuration data set before any Link record. The Options record should follow the associated Link record. The Options record is ignored.

System Action: Processing continues.

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

System Programmer Response: Move the Options record after the corresponding Link record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2161E Option name not recognized *name*

Explanation: X25IPI encountered an Options record in its X25IPI configuration data set with an unrecognized option name. The remainder of the Options record is skipped.

System Action: Processing continues.

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

System Programmer Response: Correct the option name. See *OS/390 IBM Communications Server: IP Configuration Guide* for the format of the Options record.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2162E Option packet size missing or not numeric

Explanation: X25IPI has encountered an Options statement in its *hlq*.PROFILE.TCPIP data set that has a missing or nonnumeric packet size specification. The remainder of the Options record is skipped.

System Action: Processing continues.

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

System Programmer Response: Correct the packet size.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2163E Option packet size unacceptable

Explanation: X25IPI encountered an Options record in its X25IPI configuration data set that specified an unacceptable packet size. The size should be 32, 64, 128, 256, 512, 1024, 2048, or 4096. The remainder of the Options record is skipped.

System Action: Processing continues.

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

System Programmer Response: Correct the packet size.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2164E Option window size missing or not numeric

Explanation: XNX25IPI encountered an Options statement in its X25IPI configuration data set that had a missing or nonnumeric window size. The remainder of the Options record is skipped.

System Action: Processing continues.

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

System Programmer Response: Correct the window size.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2165E Option window size not in range 1..7 or 1..127

Explanation: XNX25IPI encountered an Options statement in its X25IPI configuration data set that specified an incorrect window size. The window size should be in the range 1–7 or 1–127. The remainder of the Options record is skipped.

System Action: Processing continues.

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

System Programmer Response: Correct the window size.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2166E Option call user data missing or not hexadecimal

Explanation: The X25 server is processing an OPTION CALLDATA statement. The statement is missing the call user data, or the call user data specified was not hexadecimal.

System Action: X25 discards this record and continues.

User or Operator Response: Correct the configuration data set by correcting the CALLDATA option. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Guide* .

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTOPTN

EZB2167E Option facilities data must be hexadecimal

Explanation: The X25 server is processing an OPTION FACILITIES statement. The statement is missing the facilities data, or the facilities data specified was not hexadecimal.

System Action: X25 discards this record and continues.

User or Operator Response: Correct the configuration data set by correcting the FACILITIES option. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTOPTN

EZB2180E FAST record not preceded by a Link record

Explanation: The X25 server is processing the FAST statement in the configuration data set. The FAST statement was not preceded by a LINK statement.

System Action: XNX25IPI ignores the FAST statement.

User or Operator Response: Correct the configuration data set by either removing the FAST statement or preceding the FAST statement with a LINK statement. Rerun the TCPIPX25 catalog procedure. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Guide*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTFAST

EZB2181E FAST connect LU name prefix missing

Explanation: The X25 server is processing the FAST statement in the configuration data set. The FAST statement did not contain a VC LU prefix.

System Action: XNX25IPI ignores the FAST statement.

User or Operator Response: Correct the configuration data set by adding the prefix to the FAST statement. Rerun the TCPIPX25 catalog procedure. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Guide*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTFAST

EZB2182E FAST connect LU name too long

Explanation: The X25 server is processing the FAST statement in the configuration data set. The FAST statement contained a VC LU prefix that exceeded 8 characters.

System Action: XNX25IPI ignores the FAST statement.

User or Operator Response: Correct the configuration data set by correcting the prefix in the FAST statement. Rerun the TCPIPX25 catalog procedure. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Guide*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTFAST

EZB2183E FAST connect LU name suffix unacceptable

Explanation: The X25 server is processing the FAST statement in the configuration data set. The FAST statement contained a suffix that had an incorrect character. The suffix should be decimal or hexadecimal, as appropriate to the specified FAST connect numbering scheme.

System Action: Remaining fast-connect LU names will not be generated.

User or Operator Response: Correct the configuration data set by correcting the suffix in the FAST statement. Rerun the TCPIPX25 catalog procedure. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Guide*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTFAST

EZB2184E FAST connect LU name suffix overflow

Explanation: The X25 server is processing the FAST statement in the configuration data set. The FAST statement contains a suffix that has too many characters. The LU name suffix generated for a fast-connect virtual circuit exceeds the available field width.

System Action: Remaining fast-connect LU names will not be generated.

User or Operator Response: Correct the configuration data set by correcting the suffix in the FAST statement. Rerun the TCPIPX25 catalog procedure. For more information about the configuration data set statements, see *OS/390 IBM Communications Server: IP Configuration Guide*.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: STRTFAST

EZB2201I MCH *luname* OPNDST complete

Explanation: X25IPI has successfully established a VTAM session with the NPSI MCH LU.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: MCHOPNDC

EZB2202I MCH *luname* restart packet sent

Explanation: The X25 server has sent a restart packet to a multichannel link (MCH).

System Action: The contents of the restart packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: Use the data in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MCHOPNDC

EZB2203I MCH *lu* restarting

Explanation: An X.25 restart exchange was started on the NPSI MCH.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2205E MCH *lu* open failed

Explanation: X25IPI encountered an error opening the indicated NPSI MCH. This message is preceded by the message EZB2407E or EZB2411E reporting a VTAM error code. The MCH LU is enabled for automatic recovery by LOGAPPL, or for manual restart.

System Action: Processing continues.

User or Operator Response: Activate the NPSI MCH in VTAM, and use the X25IPI RESTART command to reacquire the MCH.

System Programmer Response: Use the VTAM error code from the EZB2407E or EZB2411E message to determine why the MCH OPNDST was unsuccessful.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2206E MCH *lu* OPNDST did not complete

Explanation: A VTAM OPNDST request on a NPSI MCH LU was posted complete, but the NIB is not marked open. The contents of the session data area (SDA) for this MCH are dumped. The MCH LU is enabled for automatic recovery by LOGAPPL, or for manual restart.

System Action: Processing continues.

User or Operator Response: Activate the NPSI MCH in VTAM, and use the X25IPI RESTART command to reacquire the MCH.

System Programmer Response: Determine the state of the NPSI MCH LU using the VTAM DISPLAY command.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2210I MCH *luname* packet level ready

Explanation: The X25 server has made a multichannel link (MCH) available for virtual circuit connections.

System Action: XNX25IPI continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: MCHOPNDC

EZB2211I MCH *luname* packet received

Explanation: The X25 server has received a packet from a multichannel link (MCH).

System Action: The contents of the packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: Use the data in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MCHRSTIP

EZB2212I MCH *lu* discarded packet during restart

Explanation: X25IPI received an incorrect packet while restarting this MCH. The incorrect packet is dumped and discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2213I MCH *lu* **restart indication, cause=***value* **diagnostic=***value*

Explanation: X25IPI received an X.25 restart indication for this MCH during the restart procedure with the indicated cause and diagnostic bytes. The X.25 network interface has been reinitialized. Virtual circuit connections on the MCH are closed, and the X.25 restart procedure is used to place the MCH back in operation. See *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning* for a list of X.25 cause and diagnostic codes. See the provider of the X.25 network service for documentation that lists additional diagnostic codes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2214I MCH *lu* **restart confirmation**

Explanation: X25IPI received a restart confirmation for this MCH. The MCH is marked ready for new connections.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2215I MCH *lu* **restart complete**

Explanation: X25IPI has completed terminating virtual circuits on an MCH undergoing restart. An X.25 restart confirmation is sent to complete the restart procedure.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2221I MCH *lu* **restart indication, cause=***cause***, diagnostic=***diagnostic*

Explanation: X25IPI received a restart indication on this MCH with the indicated cause and diagnostic bytes. The X.25 network interface has been reinitialized. Virtual circuit connections on the MCH are closed, and the X.25 restart procedure is used to place the MCH back in operation. See *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning* for a list of X.25 cause and diagnostic codes. See the provider of the X.25 network service for documentation that lists additional diagnostic codes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2222I MCH *lu* restart confirm packet sent

Explanation: X25IPI transmitted a restart confirmation packet for this MCH. The restart confirmation packet is dumped.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2223I MCH *luname* request packet sent

Explanation: The X25 server has sent a request packet from a multichannel link (MCH).

System Action: The contents of the restart packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: Use the data in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MCSTSRSR

EZB2224I MCH *lu* restarting

Explanation: X25IPI is restarting the indicated MCH.

System Action: X25IPI waits to receive a restart confirmation on the MCH.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2230I MCH *luname* packet received

Explanation: The X25 server has received a packet from a multichannel link (MCH).

System Action: The contents of the restart packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: Use the data in problem determination.

Source Data Set: XNX25IPI

Procedure Name: MCHRDYIP

EZB2231I MCH *lu* orphan packet received

Explanation: X25IPI received a packet for which it could find no associated connection. An X.25 clear request is sent on the virtual circuit.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2232I MCH *lu* diagnostic packet

Explanation: X25IPI received an X.25 diagnostic packet on this MCH. The diagnostic packet is dumped and discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the diagnostic packet to obtain additional information about X.25 network errors.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2233I MCH *lu* clear request sent

Explanation: X25IPI sent a clear request on the indicated MCH to recover from an error situation.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2234E MCH *lu* no free path available for incoming call MCH *lu* check number of logical channels on Link record

Explanation: X25IPI received an incoming call, but has no session areas available to handle it. This indicates that there are more virtual circuits subscribed than were specified on the LINK record in the X25IPI configuration data set. The incoming call is cleared.

System Action: Processing continues.

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

System Programmer Response: Increase the virtual circuit count on the Link record to match the number of virtual circuits defined in the NPSI configuration.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2235W MCH *lu* orphan packet discarded

Explanation: X25IPI received a packet that it could not associate with any connection. The packet is dumped and discarded.

System Action: Processing continues.

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

System Programmer Response: Determine the type of X.25 packet from the dump. A clear confirmation (X'17') can be discarded harmlessly.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2236E MCH *lu* unrecognized packet received

Explanation: X25IPI did not recognize the type code in a packet received from NPSI. The unrecognized packet is dumped and discarded.

System Action: Processing continues.

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

System Programmer Response: Determine the packet type from the dump; contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2250I MCH *lu* **terminating**

Explanation: X25IPI is terminating this MCH session in response to an error condition or a HALT command.

System Action: Connections on virtual circuits associated with the MCH are terminated, and the MCH session is closed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2251I MCH *lu* **closed**

Explanation: X25IPI received a CLSDST completion indication for this MCH, indicating that this MCH has closed.

System Action: The MCH LU is enabled for automatic recovery by LOGAPPL, or for manual restart.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2252I MCH *luname* **logon**

Explanation: The multichannel link (MCH) was inactive and a logon notification was received. XNX25IPI opens a VTAM session.

System Action: XNX25IPI continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: MCHRDYIP

EZB2280E MCH *lu* **session loss code** *code*

Explanation: X25IPI received the indicated MCH status change code while the MCH was not operational. See message EZB2285E for the loss codes.

System Action: The MCH LU is enabled for automatic recovery by LOGAPPL, or for manual restart.

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

System Programmer Response: Manually restart X25IPI, if necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2281I VC *vc* packet discarded on failed session

Explanation: X25IPI discarded a packet received after a session was closed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2282E MCH *lu* unexpected request completion

Explanation: X25IPI received a VTAM request completion notification for this MCH. This indicates a program error, because MCH sends are done synchronously. The notification is ignored.

System Action: Processing continues.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2283I MCH *lu* DATE error report command=*command* error=*error*

Explanation: X25IPI received a NPSI Dedicated Access to X.25 Transport Extension (DATE) error report for the indicated command and error. See *X.25 Network Control Program Packet Switching Interface Host Programming* for the error codes.

System Action: X25IPI attempts to recover from the error.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2284E MCH *lu* DATE error report command=*command* error=*error*

Explanation: X25IPI encountered a DATE error because of an incorrect logical channel number. X25IPI discards the error indication. See *X.25 Network Control Program Packet Switching Interface Host Programming* for the error codes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2285E MCH *lu* session loss code *code*

Explanation: X25IPI received a session status notification with the indicated session loss code for this MCH.

Code	Description
Less than 50	VTAM LOSTERM exit codes
50	VTAM SCIP exit UNBIND
64000	VTAM NS exit CLEANUP

64001 VTAM NS exit session initiation failure

64002 VTAM NS exit session initiation negative response

The MCH LU is enabled for automatic recovery by LOGAPPL, or for manual restart.

System Action: Processing continues.

User or Operator Response: Reactivate the NPSI MCH in VTAM. Use the X25IPI RESTART command to reacquire the MCH LU.

System Programmer Response: Use the VTAM error code to determine the reason why the session was unsuccessful.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2301I VC *ID* **incoming call from** *address* **user data** *value*

Explanation: X25IPI has received an incoming call on this virtual circuit from the indicated address with the indicated user data (protocol ID). The call is accepted or refused.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the address to determine the source of the connection.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2302I VC *vc* **call accept packet sent**

Explanation: X25IPI accepted an incoming call on this virtual circuit.

System Action: The contents of the X.25 call accept packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCINCOM

EZB2304W VC *vc* **incoming call cleared: caller not known**

Explanation: X25IPI received a call on this virtual circuit from an address not present in the Dest entries for the associated MCH. The incoming call is cleared.

System Action: Processing continues.

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

System Programmer Response: Determine the identity of the remote system initiating the call request, and add the address to the Dest list if the connection is authorized. The X.25 address of the calling system was noted in the preceding EZB2301I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2305W VC *vc* **incoming call cleared: draining**

Explanation: X25IPI cleared an incoming call because a VTAM HALT request has been issued to end communication. Incoming calls are refused once the VTAM HALT command is issued.

System Action: Processing continues.

User or Operator Response: Shutdown and restart X25IPI when VTAM is restarted.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2306E VC *vc* **incoming call cleared:** *reason*

Explanation: X25IPI cleared an incoming call on this virtual circuit because of an error in the format of the X.25 call request packet:
- Called address is not decimal.
- Calling address is not decimal.
- CUD field is too long.
- CUD field not acceptable.
- Duplicate address.
- Facilities not acceptable.
- Reverse charging has not been enabled on the associated link.
- Reverse charging refused.
- Reverse charging was specified in the call.
- Reverse charging has not been enabled on the associated link.
- The address duplicated that of another connection.

The incoming call is cleared.

System Action: Processing continues.

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

System Programmer Response: Correct the remote system that is generating the incorrect calls. The X.25 address of the calling system was noted in the preceding EZB2301I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2307I VC *vc* **clear request packet sent**

Explanation: X25IPI refused an incoming call on this virtual circuit.

System Action: The contents of the X.25 clear request packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCINCOM

EZB2308I VC *vc* **finished sending call confirm**

Explanation: A VTAM OPNDST call was accepted and the NPSI SEND completed.

System Action: XNX25IPI continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCACPTSC

EZB2310I VC *vc* outgoing call to *address*

Explanation: X25IPI is placing an outgoing call to this indicated address on this virtual circuit. Queued datagrams are sent on the connection after the call is accepted by the remote system.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the address to determine the destination of the connection.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2311I VC *vc* call request packet sent

Explanation: X25IPI placed an outgoing call on this virtual circuit.

System Action: The contents of the X.25 call request packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCLLFCL

EZB2312I VC *vc* call request sent

Explanation: X25IPI placed an outgoing call request on this virtual circuit. The X.25 NCP Packet Switching Interface (NPSI) SEND is complete.

System Action: XNX25IPI continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCALLSC

EZB2313W VC *vc* call timer expired

Explanation: The remote system has not responded to a call request within 200 seconds. The call is ended, and queued datagrams are discarded.

System Action: Processing continues.

User or Operator Response: Check the status of the remote system and the X.25 network. The X.25 address of the calling system was noted in the preceding EZB2310I message for the VC ID.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2314I VC *vc* call accepted by *address* user data *value*

Explanation: An outgoing call on this virtual circuit was accepted by the remote system at the indicated address with the indicated user data (protocol specifier). Queued datagrams are sent.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the address to determine the destination of the connection.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2315I VC *vc* **retrying call with packet size** *size*

Explanation: The packet size for this virtual circuit was reduced by the network or responder. NPSI DATE cannot handle this negotiation, thus the call is cleared and placed again with the smaller packet size.

System Action: Processing continues.

User or Operator Response: Tell the system programmer if this message recurs frequently.

System Programmer Response: Add or change an OPTIONS PACKETSIZE entry on the associated Link specifying the smallest packet size used by other systems on the X.25 network. Consider using the NPSI GATE facility rather than DATE. GATE can handle the reduced packet size without the call to be repeated.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2316E VC *vc* **outgoing call cleared:** *reason*

Explanation: X25IPI cleared an outgoing call on this virtual circuit because of an error in one of the following formats:
- The X.25 call accept packet.
- Accepting user data is too long.
- Called address is not decimal.
- Calling address is not decimal.
- Facilities not acceptable.

System Action: Processing continues.

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

System Programmer Response: Correct the remote system that is generating the incorrect calls. The X.25 address of the called system was noted in the preceding EZB2310I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2317I VC *vc* **call to** *address* **refused, cause=***cause* **diagnostic=***diagnostic*

Explanation: An outgoing call on this virtual circuit to the indicated address was refused by the X.25 network, with the indicated cause and diagnostic bytes. Datagrams queued for the remote system are discarded. A new call is attempted when the TCP acknowledgment timer expires, or when a new connection is requested to the destination. TCP connections to the destinations handled by the remote system are unsuccessful if calls are not accepted within the initial connection time-out. See *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning* for X.25 cause and diagnostic codes. See the X.25 network service provider for documentation about additional diagnostic codes.

System Action: Processing continues.

User or Operator Response: Check that the remote system and the X.25 network are operational.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2320I VC *vc* **NPSI logon LU** *lu*

Explanation: NPSI generated a VTAM session logon from the indicated LU for the virtual circuit. The NPSI LU logon is accepted.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the LU name for NPSI problem determination.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2321E VC *vc* session loss code

Explanation: X25IPI received the indicated virtual circuit (VC) status change code while the virtual circuit (VC) was not operational.

System Action: The virtual circuit (VC) LU is enabled for automatic recovery by LOGAPPL or for manual restart.

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

System Programmer Response: Restart XNX25IPI if necessary.

Source Data Set: XNX25IPI

Procedure Name: VCLOGON

EZB2322I VC *vc* OPNDST complete

Explanation: A VTAM OPNDST request on a NPSI virtual circuit (VC) LU was posted complete.

System Action: X25IPI continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCOPENC

EZB2323E VC *vc* OPNDST did not complete

Explanation: A VTAM OPNDST request on a NPSI VC LU was posted complete, but the NIB is not marked open. The contents of the session data area (SDA) for this VC are dumped. The virtual circuit call is cleared.

System Action: Processing continues.

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

System Programmer Response: Determine the state of the NPSI VC LU using the VTAM DISPLAY command.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2324I VC *vc* LU *lu* ready

Explanation: The virtual circuit LU session is ready for data transfer.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2325I VC *vc* facilities: *facilities*

Explanation: A list of facilities for this virtual circuit follows. The following are the facilities codes that can be noted:

Codes Description
pktpacket
The noted packet size is used

precedencepreced
The noted DDN precedence is applied
priority
Priority handling and charging is applied
revchg Reverse charging is applied
standard
DDN standard service is used
wdwwindow
The noted window size is used

The noted facilities are applied to the connection.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the information to determine the X.25 network facilities being used on the connection.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2326E VC *vc* **facilities field unacceptable at offset** *offset*

Explanation: X25IPI encountered an incorrect X.25 call facilities field from a remote system for this virtual circuit. The X.25 call facilities field is dumped, and the call is cleared.

System Action: Processing continues.

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

System Programmer Response: Correct the remote system that is generating the incorrect facilities. The X.25 address of the remote system was noted in the preceding EZB2301I or EZB2310I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2330I VC *vc* **call complete**

Explanation: A VTAM OPNDST request on a NPSI virtual circuit (VC) LU was posted complete.

System Action: X25IPI continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: MCHRDYIP

EZB2331I VC *vc* **data sent**

Explanation: X25IPI has sent a datagram to the X.25 NCP Packet Switching Interface (NPSI).

System Action: The contents of the X.25 data packet sequence are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCLOGON

EZB2332I VC *vc* data received

Explanation: X25IPI has received a data packet that contained an IP datagram.

System Action: The contents of the X.25 data packet sequence are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCRIPDT

EZB2333I VC *vc* packet received

Explanation: X25IPI has received a General Access To X.25 Transport Extension (GATE) control packet on this virtual circuit (VC).

System Action: The contents of the control packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCRIPDT

EZB2334E VC *vc* oversize data packet received, length=*length*

Explanation: A datagram was received on this virtual circuit that had a length exceeding the buffer size specified in the X25IPI configuration data set. Either the local or remote system is misconfigured. The packet is dumped, and the connection is cleared.

System Action: Processing continues.

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

System Programmer Response: Check the buffer size on the Buffers record in the X25IPI configuration data set. The buffer size should be large enough to hold the maximum IP datagram permitted by the usage agreements of the X.25 network. The X.25 address of the remote system was noted in the preceding EZB2301I or EZB2310I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2335E VC *vc* unrecognized packet received

Explanation: X25IPI did not recognize the type code in a packet received from NPSI GATE. The unrecognized packet is dumped and discarded.

System Action: Processing continues.

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

System Programmer Response: Determine the packet type from the dump; contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2336I VC *vc* inactivity timer expired

Explanation: The inactivity time for this virtual circuit passed with no data transferred. The virtual circuit connection is closed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2340I VC *vc* **call reset, cause=***cause* **diagnostic=***diagnostic*

Explanation: A call reset was received for this virtual circuit with the indicated cause and diagnostic bytes. The reset is confirmed, and data transfer continues. See *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning* for a list of X.25 cause and diagnostic codes. See the provider of the X.25 network service for documentation that lists additional diagnostic codes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2341I VC *vc* **reset collision**

Explanation: A reset collision occurred on this virtual circuit. Data transfer resumes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2342I VC *vc* **reset confirmed**

Explanation: A reset on this virtual circuit was confirmed by the remote system. Data transfer resumes on the virtual circuit.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2343W VC *vc* **qualified data packet discarded**

Explanation: A qualified data packet was received on an IP connection. Qualified data packets are not specified for use on IP connections. The qualified data packet is dumped and discarded.

System Action: Processing continues.

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

System Programmer Response: Correct the remote system that is sending qualified data packets. The X.25 address of the remote system was noted in the preceding EZB2301I or EZB2310I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2344W VC *vc* interrupt indication received

Explanation: An interrupt indication was received for this virtual circuit. Interrupt packets are not specified for use on IP connections. The interrupt packet is dumped, and an interrupt response is sent.

System Action: Processing continues.

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

System Programmer Response: Correct the remote system that is sending interrupt packets. The X.25 address of the remote system was noted in the preceding EZB2301I or EZB2310I message for the VC ID.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2345E VC *vc* interrupt confirmed

Explanation: An interrupt was confirmed for this virtual circuit. Interrupt packets are not specified for use on IP connections. The virtual circuit is reset, and data transfer continues.

System Action: Processing continues.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2346I VC *vc* reset confirmation packet sent

Explanation: A reset packet was received and X25IPI has sent a reset confirmation packet on this virtual circuit (VC).

System Action: The contents of the X.25 reset confirmation packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCRIPDT

EZB2347I VC *vc* interrupt confirm packet sent

Explanation: An interrupt packet was received and X25IPI has sent an interrupt confirmation packet on this virtual circuit (VC).

System Action: The contents of the X.25 interrupt confirmation packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCONFI

EZB2348I VC *vc* reset request packet sent

Explanation: X25IPI has sent a reset packet on this virtual circuit (VC) to reset the connection.

System Action: The contents of the X.25 reset request packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCONFI

EZB2350I VC *vc* **call cleared, cause=***cause* **diagnostic=***diagnostic*

Explanation: A call was cleared on this virtual circuit with the indicated cause and diagnostic bytes. A clear confirmation is sent, and the virtual circuit connection is closed. See *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning* for a list of X.25 cause and diagnostic codes. See the provider of the X.25 network service for documentation that lists additional diagnostic codes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the cause and diagnostic codes to determine the reason the call was cleared.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2351I VC *vc* **connection terminated for** *address***: sent** *count* **received** *count* **dropped** *count*

Explanation: The number of datagrams sent, received, and dropped on the virtual circuit to the remote system with the indicated X.25 address is shown when the connection is ended.

System Action: The VC LU session is closed.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2352I VC *vc* **closed**

Explanation: The call on the virtual circuit was ended. The virtual circuit is reused for new calls.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2353I VC *vc* **clear request packet sent**

Explanation: X25IPI has sent a clear packet on this virtual circuit (VC) to clear the connection. XNX25IPI is in the process of closing the connection.

System Action: The contents of the X.25 clear request packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCLROUT

EZB2354I VC *vc* **clear confirm packet sent**

Explanation: X25IPI has sent a clear confirmation packet on this virtual circuit (VC) for the clear request. XNX25IPI is in the process of closing the connection.

System Action: The contents of the X.25 clear request packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCLROUT

EZB2355I VC *vc* close pending

Explanation: A close of this virtual circuit is partially completed.

System Action: X25IPI waits for the remaining close events to occur.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2356W VC *vc* unable to clear GATE call in state P2

Explanation: X25IPI needed to clear a NPSI GATE call while the call was pending. The NPSI GATE programming interface does not allow a call to be cleared in this state.

System Action: X25IPI waits for action by the X.25 network.

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

System Programmer Response: Determine the reason the X.25 network or remote system did not respond to the call request.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2357E VC *vc* clearing limit exceeded

Explanation: No clear response was received from the remote system after 4 clear requests. The virtual circuit is marked as cleared.

System Action: Processing continues.

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

System Programmer Response: Determine the reason the X.25 network or remote system did not respond to the clear request.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2358I VC *vc* clear confirmed

Explanation: The remote system responded to a clear request on this virtual circuit. Virtual circuit termination continues; message EZB2352I should follow.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2360I VC *vc* closed LU

Explanation: X25IPI has completed a VTAM CLSDST for a vrtual circuit (VC) LU session.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCLOSRC

EZB2361W VC *vc* clearing timer expired

Explanation: The remote system did not respond to a clear request within 180 seconds. The clear request is retried 4 times.

System Action: Processing continues.

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

System Programmer Response: Determine the reason the remote system did not respond to the clear request.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2362I VC *vc* clear request packet sent

Explanation: X25IPI has sent a clear packet on this virtual circuit (VC) to clear the connection. XNX25IPI is in the process of closing the connection.

System Action: The contents of the X.25 clear request packet are displayed in subsequent messages.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCLRTMR

EZB2363E VC *vc* clearing limit exceeded

Explanation: No clear response was received from the remote system after 4 clear requests. The virtual circuit is marked as cleared.

System Action: Processing continues.

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

System Programmer Response: Determine the reason the X.25 network or remote system did not respond to the clear request.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2364I VC *vc* clear confirmation sent

Explanation: A clear request from the remote system was confirmed on this virtual circuit. Virtual circuit termination continues; message EZB2352I should follow.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2365I VC *vc* **clear sent**

Explanation: A clear request was sent on this virtual circuit. Virtual circuit termination continues; message EZB2352I should follow.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2366I VC *vc* **finished lu** *luname* **activity**

Explanation: X25IPI reports the completion of a pending VTAM request during virtual circuit (VC) closing.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VCCLWTRC

EZB2367I VC *vc* **reuse delay ended**

Explanation: The timer expired for NPSI cleanup for this virtual circuit. A new call can now be made on this virtual circuit.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2368I VC *vc* **retry busy call**

Explanation: X25IPI retries a call that previously received a busy indication from NPSI.

System Action: Processing continues.

User or Operator Response: If the error recurs, tell the system programmer about the error.

System Programmer Response: If the error recurs, check the number of virtual circuits specified on the Link record in the X25IPI configuration data set against the number defined in the NPSI configuration. Use the VTAM DISPLAY command to determine the state of the NPSI switched VC LUs.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2370E VC *vc* **inactive: VTAM request completion REQ=***request*

Explanation: A VTAM request has completed for this virtual circuit after the connection has ended.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2371E VC *vc* inactive: session loss code *code*

Explanation: X25IPI received a session status notification with the indicated session loss code for this virtual circuit after the connection was ended. The notification is ignored. See message EZB2383I for the loss codes.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2372I VC *vc* inactive: session cleanup

Explanation: Session cleanup is in progress for this inactive virtual circuit. The VC LU session is closed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2373E VC *vc* inactive: VTAM logon refused

Explanation: A late logon from a NPSI VC LU was refused after a call was cleared. The logon is rejected.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2374I VC *vc* packet discarded on dead connection

Explanation: A packet was received on this virtual circuit after the connection was ended. The packet is discarded.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2381E VC *vc* DATE error report command=*command* error=*error*

Explanation: X25IPI received a NPSI DATE error report for the indicated command and error. See *X.25 Network Control Program Packet Switching Interface Host Programming* for the error codes.

System Action: X25IPI attempts to recover from the error.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2382E VC *vc* **GATE error report command=***command* **error=***error*

Explanation: X25IPI received a NPSI GATE error report for the indicated command and error. See *X.25 Network Control Program Packet Switching Interface Host Programming* for the error codes.

System Action: X25IPI attempts to recover from the error.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2383E VC *vc* **session loss code** *value* **on LU** *lu*

Explanation: A session loss status notification was received for this virtual circuit with the indicated loss code.

Code	Description
Less than 50	VTAM LOSTERM exit codes
50	VTAM SCIP exit UNBIND
64000	VTAM NS exit CLEANUP
64001	VTAM NS exit session initiation failure
64002	VTAM NS exit session initiation negative response

The virtual circuit connection is closed.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Use the VTAM error code to determine why the session was unsuccessful.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2384I VC *vc* **discarded packet** *packet* **in state** *state*

Explanation: The indicated packet was discarded on this virtual circuit to accomplish error recovery. Virtual circuit error recovery is completed.

System Action: Processing continues.

User or Operator Response: See message EZB2021I VC for the call state codes.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2385E VC *vc* **received packet** *packet* **invalid in state** *state*

Explanation: The indicated received packet was incorrect for the current virtual circuit state. The virtual circuit is reset or cleared.

System Action: Processing continues.

User or Operator Response: See message EZB2021I VC for the call state codes. Tell the system programmer about the error.

System Programmer Response: Determine the reason the X.25 network or remote system sent the incorrect packet.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2386E VC *vc* **discarded packet** *packet* **in state** *state*

Explanation: The indicated packet for VC was discarded because the state was not valid for the current virtual circuit state. The virtual circuit *vc* is cleared.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: Determine the reason the remote X.25 network sent the incorrect *state* value for the circuit.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2401E VTAM GENCB failed, R15=*value* **R0=***value*

Explanation: A VTAM GENCB call was unsuccessful with the indicated R15 and R0 values.

System Action: X25IPI does not start.

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

System Programmer Response: Reassemble X25IPI with the most recent VTAM macro library.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2402E VTAM ACB OPEN failed, R15=*value* **ACBERFLG=***value*

Explanation: A VTAM OPEN request failed with the indicated R15 and ACBERFLG values.

System Action: X25IPI does not start.

User or Operator Response: Activate the X25IPI application in VTAM.

System Programmer Response: Use the VTAM error codes to determine why the VTAM ACB OPEN macro was unsuccessful. Refer to *OS/390 IBM Communications Server: SNA Programming* where codes can be found.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2403E VTAM SETLOGON failed, RTNCD=*rc* **FDB2=***value*

Explanation: A VTAM SETLOGON request failed with the indicated return code and FDB2 values.

System Action: Processing continues, but virtual circuit (VC) LU logons will fail.

User or Operator Response: None.

System Programmer Response: Use the VTAM error code to determine the reason for the failure of the VTAM SETLOGON request.

Source Data Set: XNX25IPI

Procedure Name: VTAMENAB

EZB2404E VTAM SETLOGON QUIESCE failed, RTNCD=*rc* FDB2=*value*

Explanation: A VTAM SETLOGON QUIESCE request failed with the indicated return code and FDB2 values.

System Action: X25IPI termination continues.

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

System Programmer Response: Use the VTAM error codes to determine why the VTAM SETLOGON macro. Refer to the *OS/390 IBM Communications Server: SNA Programming* where codes can be found.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2405E VTAM ACB CLOSE failed, R15=*value* ACBERFLG=*value*

Explanation: A VTAM CLOSE request failed with the indicated return values.

System Action: X25IPI termination continues.

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

System Programmer Response: Use the VTAM error codes to determine why the VTAM CLOSE macro was unsuccessful. Refer to the *OS/390 IBM Communications Server: SNA Programming* where codes can be found.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2406E VTAM GENCB failed, R15=*value* R0=*value*

Explanation: A VTAM GENCB call failed with the indicated R15 and R0 values.

System Action: The X25 server continues processing.

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

System Programmer Response: Reassemble X25IPI with the most recent VTAM macro library.

Source Data Set: XNX25IPI

Procedure Name: VTAMSIN

EZB2407E VTAM *VTAM request lu* failed, RTNCD=*rc* FDB2=*value*

Explanation: The indicated VTAM request was unsuccessful with the indicated return code and FDB2 values. The virtual circuit can become unusable.

System Action: Processing continues.

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

System Programmer Response: Use the VTAM error codes to determine why the VTAM request was unsuccessful. Refer to the *OS/390 IBM Communications Server: SNA Programming* where codes can be found. Determine the state of the NPSI LU using the VTAM DISPLAY command. If all virtual circuits become unusable, HALT and restart the X25IPI application.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2410I VTAM request complete for *luname* REQ=*value*

Explanation: X25IPI completed a VTAM request. See message EZB2411E for the request codes.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: VTAMCHK

EZB2411E VTAM request failed for *lu* **REQ=***request* **RNTCD=***rc* **FDB2=***value* **sense=***value* **(***caller***)**

Explanation: The indicated VTAM request was unsuccessful with the indicated return code, FDB2, and sense values. The caller field specifies the routine that issued the request. The following are the request codes:

Request Code	Description
X'17'	OPNDST
X'1F'	CLSDST
X'22'	SEND
X'23'	RECEIVE

System Action: If the error occurred on a NPSI MCH session, the MCH is shut down. If the error occurred on a VC session, the connection is closed.

User or Operator Response: If the error occurred on a NPSI MCH session, activate the NPSI MCH in VTAM and use the X25IPI RESTART command to reacquire the MCH.

System Programmer Response: Use the VTAM error codes to determine why the VTAM ACB OPEN macro was unsuccessful. Refer to the *OS/390 IBM Communications Server: SNA Programming* where codes can be found.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2420I Logon exit refusing session

Explanation: XNX25IPI is refusing a logon session because:

- For an MCH with LOGAPPL coded, the MCH is not defined by a LINK entry, or the MCH is not in a state where a logon is expected
- The logon was not initiated by X.25 NPSI
- A virtual circuit is not in a state where a logon is expected, possibly because a call request was unsuccessful.

The NPSI LU session logon is refused.

System Action: Processing continues.

User or Operator Response: Notify the system programmer if the message is issued frequently, and connections are failing.

System Programmer Response: Check the source of the logons; determine why the call request was unsuccessful.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2421E Network services exit: Unrecognized request type

Explanation: XNX25IPI received an unrecognized Network Services RU from VTAM. The following are the supported RU:

- Cleanup Session (X'810629')
- Notify (X'810620')
- NS Procedure Error (X'010604').

VTAM is notified that the NS RU was not handled.

System Action: Processing continues.

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

System Programmer Response: Obtain a VTAM buffer trace to determine the Network Services RU type.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2422I Network services exit: Cleanup session notification posted

Explanation: A VTAM Cleanup Session RU was received. The VTAM session is ended. A failure message for the MCH or virtual circuit (VC) session follows.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2423W Network services exit: Session initiation failure notification posted

Explanation: A VTAM Session initiation failure RU was received. The VTAM session is ended. A failure message for the MCH or VC session can follow.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2424T Session control exit: Unrecognized or unexpected request type

Explanation: An unexpected VTAM Session Control RU was received. ABEND code X'941' is displayed with this message.

System Action: X25IPI abends.

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

System Programmer Response: Determine the source of VTAM session control request. Submit the ABEND dump to the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2425I Session control exit: UNBIND notification posted

Explanation: A VTAM Session Control RU of type UNBIND was received. The VTAM session is ended. A failure message for the MCH or VC session can follow.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2431E VTAM terminating, reason unknown

Explanation: A VTAM Shutdown notification was received. The cause is unknown. This rules out a standard shutdown, HALT QUICK, INACT, or HALT CANCEL reason codes.

System Action: X25IPI ends.

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

System Programmer Response: Determine the source of the Shutdown notification.

Source Data Set: NX25IPI

Procedure Name: None.

EZB2432I VTAM HALT issued, drain flag set

Explanation: A VTAM HALT request was issued.

System Action: X25IPI ends after current connections are closed. New connections are refused.

User or Operator Response: Restart X25IPI, if appropriate, after VTAM resumes.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2433I VTAM HALT QUICK or VARY INACT, *application* issued

Explanation: A VTAM HALT QUICK or VARY INACT request was issued.

System Action: X25IPI ends.

User or Operator Response: Restart X25IPI, if appropriate, after VTAM resumes.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2434I VTAM HALT CANCEL issued

Explanation: A VTAM HALT CANCEL request was issued.

System Action: X25IPI ends.

User or Operator Response: Restart X25IPI, if appropriate, after VTAM resumes.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2453I IP *AS_userid* path refused for userid *userid*, draining

Explanation: A DLC connection from the TCPIP address space was refused because a VTAM HALT command had been issued.

System Action: Processing continues.

User or Operator Response: HALT and restart the X25IPI application when VTAM is restarted.

System Programmer Response: Assist the user as necessary.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2460E Destination address *address* not configured

Explanation: An IP datagram was received from TCPIP for transmission to the indicated address, which is not listed in the Dest entries in the X25IPI configuration data set. The IP datagram is discarded.

System Action: Processing continues.

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

System Programmer Response: Add a Dest record to the X25IPI configuration data set for the IP address or network. Check the GATEWAY entries in *hlq*.PROFILE.TCPIP for a misrouted network number.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2462E IP *AS_userid* **rejected message: too long IP** *AS_userid* **rejected message: too short**

Explanation: A DLC message was received from the TCPIP address space which was too long or too short. The message is rejected.

System Action: Processing continues.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2480I IP *AS_id* **disconnected: sent** *count* **received** *count* **dropped** *count*

Explanation: The number of datagrams sent, received, and dropped on the DLC connection to the TCPIP address space are shown when the connection is terminated.

System Action: X25IPI ends.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2491I DLC AS *asname* **path severed**

Explanation: The DLC connection to TCPIP has been severed, most likely because TCPIP is being terminated or the X25NPSI device has been STOPped.

System Action: X25IPI exits.

User or Operator Response: Restart X25IPI, if appropriate, after TCPIP is restarted.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: IPRDYPN

EZB2492I DLC *AS_path***accepted for userid** *userid*

Explanation: A DLC connection from the TCPIP address space was accepted. IP datagrams are transferred.

System Action: Processing continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2493I DLC received data

Explanation: X25IPI displays the contents of an IP datagram received from TCPIP in subsequent messages.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: IPRDYPN

EZB2494I DLC send datagram length *length* **next offset** *offset*

Explanation: X25IPI displays the length of the datagram to be sent to TCPIP.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: IPSENDDG

EZB2495I DLC send total length *length*

Explanation: X25IPI displays the total length of IP datagrams to be sent to TCPIP.

System Action: The X25 server continues processing.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: XNX25IPI

Procedure Name: IPSENDDG

EZB2496E XNX25IUT failed: CSM storage problem

Explanation: XNX25IUT interface returned a return code of 4. This indicates an error with CSM storage.

System Action: X25IPI closes connection with TCPIP.

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

System Programmer Response: Check for correct installation of the MVS TCPIP product.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2497E DLC *function* **failed: STAFD=** *reason*

Explanation: XNX25IUT interface returned a return code of 8. The indicated DLC function, issued by XNX25IPI, failed for the reason indicated in the STAFD reason field.

1 INIT

2 SEND

3 RECV

4 CLEAR

5 TERM

System Action: The DLC connection is severed.

User or Operator Response: Tell the system programmer.

System Programmer Response: Perform the action described in *OS/390 IBM Communications Server: IP and SNA Codes* for the indicated status code.

Source Data Set: XNX25IPI

Procedure Name: None.

EZB2498E XNX25IUT failed: Unexpected return code

Explanation: XNX25IUT interface returned a return code other than 0, 4, or 8.

System Action: Processing continues.

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

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: XNX25IPI

Procedure Name: None.

Chapter 5. EZB2500—EZB2785

Kerberos Messages

This section contains Kerberos messages.

EZB2500E opening database *Kerberos data set name*: *error message*

Explanation: An error occurred when the Kerberos system data set opened. The error message specifies the type of error that occurred.

System Action: The Administration Server halts. TCPIP continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Make the database accessible to the Kerberos or ADM@SRV server. If the database does not exist, run the KDB@INIT program to create it. For more information about configuring the Kerberos server, see *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: ADM@SERV

Procedure Name: Main

EZB2501I Usage: admin@server [-h] [-n] [-r realm] [-d dbname][-f filename]

Explanation: This is the help text for the ADMIN@SERVER command. For more information about the ADMIN@SERVER COMMAND, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ADM@SERV

Procedure Name: main

EZB2502I Unable to get local realm. Fix krb.conf or use -r.

Explanation: The Kerberos Administration Server (KAS) is unable to get the realm name. The realm is the name of a local administrative entity; it identifies each independent Kerberos site.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: As indicated in the message text, 2 options exist. Either add the correct realm name (commonly the domain name) to the first line of your KRB.CONF data set, or use the -r parameter to the ADM@SERV procedure to have the Kerberos Authentication System prompt you for the Kerberos realm. For more information about the KRB.CONF data set, see *OS/390 IBM Communications Server: IP Configuration Reference*. For more information about the KINIT command, see *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: ADM@SERV

Procedure Name: main

EZB2503I KADM Server *server name* **initializing**

Explanation: The Kerberos Administration Server for the local realm is starting.

System Action: The Kerberos Administration Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ADM@SERV

Procedure Name: main

EZB2504E error: kerb_init( ) failed

EZB2505E error: *error message*

Explanation: The Kerberos Administration Server for the local realm failed to initialize.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Verify that the MVSKERB and ADM@SRV cataloged procedures in TCPIP are correct, updated, and include the proper DD statements, as required. For details, see *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: ADM@SERV

Procedure Name: main

EZB2506I Admin Server (kadm server) has completed operation.

Explanation: The Kerberos Administration Server (KAS) has shut down.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: ADM@SERV

Procedure Name: close_syslog()

EZB2507I Info in Database for *principal name.instance*

Explanation: The Kerberos Administration Server (KAS) has found the principal name and instance in the Kerberos database.

System Action: Kerberos and the Kerberos Administration Server continue.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KA@SUPP

Procedure Name: prin_vals

EZB2508I Max Life: *life* **Exp Date:** *date*

Explanation: This message specifies the maximum ticket lifetime and the resulting expiration date and time for the ticket. The default lifetime of a Kerberos ticket is 8 hours.

System Action: Kerberos and the Kerberos Administration Server continue.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KA@SUPP

Procedure Name: prin_vals

EZB2509I **Attribs:** *hex number* **key:** *low high*

Explanation: This message indicates the values for the attributes and the low and high values for outstanding keys for the user. Attributes are used to store the user's ticket information. The keys are random numbers used in the encryption process.

System Action: Kerberos and the Kerberos Administration Server continue.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KA@SUPP

Procedure Name: prin_vals

EZB2510I **Kerberos admin server loses.....** *error messages*

Explanation: This message shows the error messages for any user information lost from the Kerberos database.

System Action: Kerberos exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use the KDB@UTIL command to examine and edit the contents of the Kerberos database.

Source Data Set: KA@SUPP

Procedure Name: prin_vals

EZB2511I **Welcome to the Kerberos Administration Program, version 2**

EZB2512I **Type "help" if you need it.**

Explanation: This initial welcome message is followed by a message that prompts you for the Kerberos help facility. They are followed by the "admin:" prompt.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Type `help` if you need information about using the Kerberos Database Manager (KDBM).

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: do_init

EZB2514I **kerberos error:** *Kerberos name*

Explanation: The Kerberos ID you entered at the "userid" prompt after the KADMIN command is incorrect or is not found in the ADM@SRV.ADM@ACL data sets:

ADM@SRV.ADM@ACL.ADD
ADM@SRV.ADM@ACL.GET
ADM@SRV.ADM@ACL.MOD

System Action: The Kerberos Administration Server continues.

User or Operator Response: Enter the correct Kerberos ID. If that is unsuccessful, use KDB@EDIT to register user IDs with the Kerberos database. Your Kerberos name must be included in the following data sets:

ADM@SRV.ADM@ACL.ADD
ADM@SRV.ADM@ACL.GET
ADM@SRV.ADM@ACL.MOD

in order to perform the corresponding database operations. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: setvals

EZB2515E kadm error for realm *realm name*: *error message*

Explanation: The principal name you entered at the "userid" prompt after the KADMIN command did not correspond to a realm name in the appropriate ADM@SRV.ADM@ACL data sets.

System Action: The Kerberos Administration Server halts.

User or Operator Response: Enter the correct Kerberos ID. If that is unsuccessful, use KDB@EDIT from the ADM@SRV user ID to register with the Kerberos database. The Kerberos instance must be `admin`. Your Kerberos realm must be included in the following data sets

ADM@SRV.ADM@ACL.ADD
ADM@SRV.ADM@ACL.GET
ADM@SRV.ADM@ACL.MOD

in order to perform the corresponding database operations. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: setvals

EZB2516I Usage: change_password loginname

Explanation: You entered an incorrect CHANGE_PASSWORD (CPW) subcommand. This message shows the correct syntax for the subcommand.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Correct the change_password subcommand and wait for the `New password` prompt from Kerberos.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_password

EZB2517I Password changed for *principal name*.

Explanation: The CHANGE_PASSWORD subcommand was successful.

System Action: The Kerberos Administration Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_password

EZB2518I kadmin: *error message* **while changing password for** *principal name*

Explanation: The new password that you entered at the prompt cannot be accepted as your new Kerberos password for the reason indicated.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Try the CHANGE_PASSWORD subcommand again, using a different password as indicated by the error message. For guidelines on passwords, type HELP CPW.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_password

EZB2519E Error reading password; password unchanged

Explanation: The new password could not be read by KDB@SERV.C because it was of an incorrect length.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Try the CHANGE_PASSWORD subcommand again, using a different password length. For more guidelines on passwords, type HELP CPW.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_password

EZB2520E kadmin: Principal does not exist.

Explanation: The Kerberos Administration Server cannot find the principal name that you entered as an argument to the CHANGE_PASSWORD subcommand. The principal name is not in the Kerberos database.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Enter the correct Kerberos principal. If that is unsuccessful, use KDB@EDIT from the ADM@SRV user ID to register with the Kerberos database. Your Kerberos name must be included in the following data sets:

ADM@SRV.ADM@ACL.ADD
ADM@SRV.ADM@ACL.GET
ADM@SRV.ADM@ACL.MOD

in order to perform the corresponding database operations. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_password

EZB2521E Usage: change_admin_password

Explanation: You entered an incorrect CHANGE_ADMIN_PASSWORD subcommand. This message shows the correct syntax for the subcommand, which requires no arguments.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Correctly enter the CHANGE_ADMIN_PASSWORD subcommand and wait for the `New password` prompt from Kerberos.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_admin_password

EZB2522I Admin password changed

Explanation: The CHANGE_ADMIN_PASSWORD procedure was successful.

System Action: The Kerberos Administration Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_admin_password

EZB2523E kadm error: *error message*

Explanation: The new ADMIN instance password that you entered at the prompt cannot be accepted as your new password for the reason indicated.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Try the CHANGE_ADMIN_PASSWORD subcommand again, using a different password. For more guidelines on passwords, type HELP CAP.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_admin_password

EZB2524E Error reading password; password unchanged

Explanation: The new ADMIN instance password could not be read by the Kerberos server.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Try the CHANGE_ADMIN_PASSWORD subcommand again, using a different password. For more guidelines on passwords, type HELP CAP.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: change_admin_password

EZB2525I Usage: add_new_key user_name.

Explanation: You have entered the ADD_NEW_KEY subcommand to register a new principal name with the Kerberos database. This message is followed by another message that prompts you for a password.

System Action: The Kerberos Administration Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: add_new_key

EZB2526I *principal name* **added to database.**

Explanation: The principal name and its corresponding password have been added to the Kerberos database. The ADD_NEW_KEY subcommand was successful.

System Action: The Kerberos Administration Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: add_new_key

EZB2527E kadm error: *error message*

Explanation: The new principal name could not be added to the Kerberos database for the reason indicated in the error message.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Correct the new principal name, and reenter the ADD_NEW_KEY subcommand. For more information, type HELP ANK.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: add_new_key

EZB2528E Error reading password; *principal name* **not added**

Explanation: The password you entered for the new principal name is incorrect.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Reissue the ADD_NEW_KEY subcommand, making sure the password is correct. For more information, type HELP ANK.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: add_new_key

EZB2529I kadmin: Principal already exists.

Explanation: You entered a principal name that is already registered in the Kerberos database. The Kerberos server is configured so as to prohibit duplicate principal names, so it must reject your request.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Reissue the ADD_NEW_KEY subcommand, using a different principal name.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: add_new_key

EZB2530I Usage: get_entry username

Explanation: You entered the GET_ENTRY subcommand at the “admin” prompt. This message precedes a password prompt.

System Action: The Kerberos Administration Server continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: get_entry

EZB2531E kadm error: *error message*

Explanation: An error occurred in the Kerberos database operation, as indicated. The server could not gain entry into the database.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Reissue the GET_ENTRY subcommand, making sure to correct the error as indicated. For more information, type HELP GET.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: get_entry

EZB2532I kadmin: Principal does not exist.

Explanation: No entry exists in the Kerberos database for the principal name.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Enter the correct Kerberos ID. If that is unsuccessful, use KDB@EDIT from the ADM@SRV user ID to register with the Kerberos database. The Kerberos instance must be `admin`. Your Kerberos name must be included in the following data sets

ADM@SRV.ADM@ACL.ADD
ADM@SRV.ADM@ACL.GET
ADM@SRV.ADM@ACL.MOD

in order to perform the corresponding database operations. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the user as required.

Source Data Set: KADMIN

Procedure Name: get_entry

EZB2533I Welcome to the Kerberos administration program

EZB2534I Type "?" to get

EZB2535I a list of requests that are available. You can

EZB2536I get help on each of

EZB2537I the commands by typing "help command_name".

EZB2538I Some functions of this

EZB2539I program will require an "admin" password

EZB2540I from you. This is a password

EZB2541I private to you, that is used to authenticate

EZB2542I requests from this

EZB2543I program. You can change this password with

EZB2544I the "change_admin_password"

EZB2545I (or short form "cap") command. Good Luck!

Explanation: These are the introductory messages that appear when you type `help` at the "admin:" prompt. They briefly explain how the Kerberos help facility works.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Type `?` or `help` followed by the subcommand that you want to learn about.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2546I **Usage: change_password user_name.**

EZB2548I **user_name is the name of the user whose password**

EZB2549I **you wish to change.**

EZB2550I **His/her password is changed in the kerberos database**

EZB2551I **When this command is issued, first the "Admin"**

EZB2552I **password will be prompted**

EZB2553I **for and if correct the user's new password will**

EZB2554I **be prompted for (twice with**

EZB2555I **appropriate comparison). Note: No minimum password**

EZB2556I **length restrictions apply, but**

EZB2557I **longer passwords are more secure.**

Explanation: These are the help messages that appear when you type `help` followed by the CHANGE_PASSWORD (or CPW) subcommand at the "admin:" prompt.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Use the information provided here to assist you in changing your Kerberos password.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2558I **Usage: change_admin_password.**

EZB2560I **This command takes no arguments and is used**

EZB2561I **to change your private**

EZB2562I **"Admin" password. It will first prompt for**

EZB2563I **the (current) "Admin"**

EZB2564I **password and then ask for the new password**

EZB2565I **by prompting:**

EZB2567I New password for <Your User Name>.admin:

EZB2569I Enter the new admin password that you desire

EZB2570I (it will be asked for

EZB2571I twice to avoid errors).

Explanation: These are the help messages that appear when you type `help` followed by CHANGE_ADMIN_PASSWORD (or CAP) at the "admin:" prompt.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Use the information supplied here to help you change your ADMIN instance password.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2572I Usage: add_new_key user_name.

EZB2574I user_name is the name of a new user to put

EZB2575I in the kerberos database. Your

EZB2576I "Admin" password and the user's password

EZB2577I are prompted for. The user's

EZB2578I password will be asked for

EZB2579I twice to avoid errors.

Explanation: These are the Kerberos help messages that appear when you type `help` followed by ADD_NEW_KEY (or ANK) at the "admin:" prompt.

System Action: Kerberos continues.

User or Operator Response: Use the information supplied here to assist you in registering your new principal name with the Kerberos database.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2580I Usage: get_entry user_name.

EZB2582I user_name is the name of a user whose

EZB2583I entry you wish to review. Your

EZB2584I "Admin password is prompted for.

EZB2585I The key field is not filled in, for

EZB2586I security reasons.

Explanation: These help messages appear when you type `help` followed by GET_ENTRY (or GET) at the "admin:" prompt.

System Action: Kerberos continues.

User or Operator Response: Use the information supplied in these messages to assist you in getting the entry.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2587I Usage: list_requests

EZB2589I This command lists what other commands are

EZB2590I currently available.

Explanation: These are the help messages that appear when you type `help` followed by LIST_REQUESTS (or LR or ?) at the "admin:" prompt.

System Action: Kerberos continues.

User or Operator Response: Use the information provided in these messages to assist you in displaying a list of possible subcommands for the KADMIN command.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2591I Usage: quit

EZB2593I This command exits this program.

Explanation: These help messages appear when you type `help`, followed by `quit`, `exit`, or `q` at the "Admin:" prompt.

System Action: Kerberos continues.

User or Operator Response: Use the help messages to assist you in learning how to exit the Kerberos program.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2594I Sorry there is no such command as *command*. type "help" for more information.

Explanation: You have entered an incorrect help subcommand.

System Action: Kerberos continues.

User or Operator Response: Correct the help subcommand as necessary, and reissue it.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: help

EZB2595E *command*: *error message*

Explanation: The KADMIN command you attempted to enter was incorrect. It is shown, along with an error message that explains the problem.

System Action: KADMIN exits.

User or Operator Response: Reinitialize the Kerberos server using the KINIT command. Then reissue the corrected KADMIN command. See messages EZB2596I–EZB2600I.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: go_home

EZB2596I Usage: kadmin [-u admin_name] [-r default_realm]

Explanation: This message shows the syntax and parameters for the KADMIN command. See also messages EZB2598I–EZB2600I.

System Action: Kerberos continues.

User or Operator Response: For more information about the KADMIN command, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: go_home

EZB2598I -m allows multiple admin requests to be

EZB2599I serviced with one entry of admin

EZB2600I password.

Explanation: You can add the -m parameter to the KADMIN command to perform multiple kadmin operations without having to reenter your password for each operation.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: go_home

EZB2601I Your userid:

Explanation: This message is the prompt for the administrator's principal_name.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Enter your administrator's principal_name.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: do_init

EZB2602I Error reading admin password.

Explanation: The ADMIN instance password that you entered is incorrect.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: Enter a correct password at the prompt.

System Programmer Response: Assist the user as necessary.

Source Data Set: KADMIN

Procedure Name: get_admin_password()

EZB2603E Principal *userid*.admin@*realm* does not exist.

Explanation: The principal name is not registered with the Kerberos database.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: If you mistyped the principal name, enter the correct principal name and try again. If the ADMIN principal does not exist, use KDB@EDIT to register the ADMIN principal.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: get_admin_password()

EZB2604I Incorrect admin password.

Explanation: The ADMIN instance password that you entered is incorrect.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: Enter a correct password at the prompt.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: get_admin_password()

EZB2605I Cleaning up and exiting.

Explanation: The ADMIN instance password that you entered was not registered with the Kerberos database, or was incorrect. For security reasons, the Kerberos database closes.

System Action: The Kerberos Administration Server halts. TCPIP continues.

User or Operator Response: Retry the KADMIN command, making sure to enter the correct password.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: get_admin_password()

EZB2606I Null passwords are not allowed; try again.

Explanation: The principal has registered with a null password. Null passwords are not accepted when registering with the Kerberos database.

System Action: The Kerberos Administration Server continues.

User or Operator Response: Use the CHANGE_ADMIN_PASSWORD command to change the password to a non-null string.

System Programmer Response: None.

Source Data Set: KADMIN

Procedure Name: get_password

EZB2608E *error message*

Explanation: An error occurred while the Kerberos Administration Server was getting the password tickets.

System Action: TCPIP continues.

User or Operator Response: Use the error message to help diagnose the problem. Reinitialize the Kerberos server and reissue the KADMIN command.

System Programmer Response: None.

Source Data Set: PERROR

Procedure Name: com_err

EZB2609I **Available** *subsystem name* **requests:** *subcommands*

Explanation: If you enter the LIST_REQUESTS subcommand at the “admin:” prompt, this message displays a list of available subcommands for the particular subsystem.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: LIST@REQ

Procedure Name: ss_list_requests

EZB2610I **Usage: %s et_file [-n]**

Explanation: This message shows the syntax for the database debugging routine, et_file. If debugging is enabled, this routine compiles a table of incorrect database entries.

System Action: Kerberos exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SS@ER@ET

Procedure Name: main

EZB2611I **Line number** *number*; **last token was** ‘*token name*’

Explanation: When you issue the GET_ENTRY subcommand, this message provides the line number and last token of the Kerberos database entry.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: SS@ER@ET

Procedure Name: yyerror

EZB2615E *key data set*: **bad instance name:** *name*

Explanation: The instance name that you entered at the prompt is incorrect. It does not correspond to any of the instance names for the key data set specified.

System Action: The EXT@SRVT command exits.

User or Operator Response: Enter a correct instance name for the key data set. For more information about generating key data sets, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2616E Couldn't read master key.

Explanation: The master key that you entered at the prompt was incorrect.

System Action: The EXT@SRVT command exits.

User or Operator Response: Enter the correct Kerberos master key (password) at the system prompt.

System Programmer Response: None.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2617E *host name*: **couldn't get local realm**

Explanation: The local realm definition is missing from the first line of the *user_id*.KRB.CONF data set.

System Action: The EXT@SRVT command exits.

User or Operator Response: Add the local realm name to the first line of your *user_id*.KRB.CONF data set. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2618E Couldn't create file '*data set***'.**

Explanation: The EXT@SRVT program cannot generate the *userid*.INST.SRVTAB data set for the specified instance name.

System Action: The EXT@SRVT command exits.

User or Operator Response: None.

System Programmer Response: Verify that the ADM@SRV address space has read/write access to the Kerberos database. For more information about generating key data sets, see *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2619I Generating '*data set***'....**

Explanation: The system is generating the ADM@SRV.INST.SRVTAB data set for the specified instance.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2620I More than 40 found...

Explanation: More than 40 entries already exist in the Kerberos database that share the specified principal name. This exceeds the limit for multiple instances.

System Action: The EXT@SRVT command exits.

User or Operator Response: None.

System Programmer Response: Use the KDB@UTIL command to remove some of the instances.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2621I Error writing output file. Terminating.

Explanation: The program encountered errors writing entries to the *userid*.INST.SRVTAB key data set.

System Action: EXT@SRVT exits.

User or Operator Response: Make sure sufficient disk space exists for the newly-created data sets, or use the appropriate I/O support functions under MVS to find the cause of the output error.

System Programmer Response: None.

Source Data Set: EXT@SRVT

Procedure Name: FWrite

EZB2622E Usage: ext_srvtab [-n] [-r realm] instance [instance ...]

Explanation: If the program cannot parse the command line, this message displays the proper syntax for the EXT@SRVT command.

System Action: EXT@SRVT exits.

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

System Programmer Response: None.

Source Data Set: EXT@SRVT

Procedure Name: usage

EZB2623I Did not find any principles. '*data set*' was not generated

Explanation: No entries were found in the Kerberos database for the principal, so no instance names could be associated with them.

System Action: Kerberos continues.

User or Operator Response: Ask the Kerberos administrator to verify that your principal name is included in the Kerberos data base.

System Programmer Response: Assist the user as necessary.

Source Data Set: EXT@SRVT

Procedure Name: main

EZB2625I You are about to destroy the Kerberos database on this machine.

EZB2626I Are you sure you want to do this (y/n)?

Explanation: You have entered the KDB@DEST command, which erases the Kerberos database data sets for your host.

System Action: The KRB@DEST command continues.

User or Operator Response: If you want to erase the ADM@SRV.PRINCPL.DAT and ADM@SRV.PRINCIPL.IDX data sets, type y. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@DEST

Procedure Name: main

EZB2627I Database deleted at *data set*

Explanation: The ADM@SRV.PRINCIPL.DAT and ADM@SRV.PRINCIPL.IDX data sets have been erased.

System Action: The KRB@DEST command continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@DEST

Procedure Name: main

EZB2628I Database cannot be deleted at *data set*

Explanation: You do not have authorization to delete the Kerberos database on this host.

System Action: The KRB@DEST command continues.

User or Operator Response: Issue the KDB@DEST command from the ADM@SRV ID.

System Programmer Response: Assist the user as necessary.

Source Data Set: KDB@DEST

Procedure Name: main

EZB2629I Database not deleted.

Explanation: If you responded n to the prompt (see message EZB2626I), this message confirms that the database was not deleted.

System Action: The KRB@DEST command continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@DEST

Procedure Name: main

EZB2630E *program name*: **size of long is 4.**

Explanation: The maximum number of digits you can enter for any of the KDB@EDIT variables is 4.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2631E *program name*: **illegal flag "***flag***"**

Explanation: The calling program contains an incorrect flag.

System Action: KDB@EDIT exits.

User or Operator Response: Correct the offending flag. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2632I Opening database...

Explanation: After you enter the Kerberos user name, this message indicates that the program is accessing the database.

System Action: KDB@EDIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2633E Could not open altername database name

Explanation: The Kerberos master database does not exist or is locked.

System Action: KDB@EDIT exits.

User or Operator Response: Use the KDB@INIT command to create the Kerberos database. Also make sure that the ADM@SRV ID is allocated. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2634E Couldn't read master key.

Explanation: The KDB@EDIT program could not access the Kerberos database because the master key you entered was incorrect.

System Action: KDB@EDIT exits.

User or Operator Response: Enter the valid Kerberos master key. See the system administrator for assistance.

System Programmer Response: Assist the user as necessary.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2635E *program name*: **Kerberos error on default value lookup,** *principals* **found.**

Explanation: The KDB@EDIT program found more than one instance of the principal name in the database.

System Action: KDB@EDIT exits.

User or Operator Response: Delete the duplicate principle from the Kerberos data base.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2636I Previous or default values are in [brackets] ,

EZB2637I enter return to leave the same, or new value.

Explanation: The Kerberos database entries that you can change are in brackets. Values shown are the default or previously entered values.

System Action: KDB@EDIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: main

EZB2638I Principal name: *username*

Explanation: This is the principal user name or service name. If a principal name already exists in the Kerberos database, it is shown here.

System Action: KDB@EDIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2639I Instance: *instance name*

Explanation: If the instance name already exists in the Kerberos database, this message displays it.

System Action: KDB@EDIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2640I <Not found>, Create [y] ?

Explanation: If no instance name exists in the Kerberos database, this message prompts you to enter one.

System Action: KDB@EDIT continues.

User or Operator Response: Typing `y` or `Y`, or pressing the space bar and then the ENTER key to enter a null character causes Kerberos to continue. Entering any other character causes Kerberos to exit.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2641I Principal: *user name*, **Instance:** *instance*, **kdc_key_ver:** *version number*

Explanation: This message confirms that the principal has been created in the Kerberos database and specifies the master key version number.

System Action: KDB@EDIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2642I Change password [n] ?

Explanation: This message prompts you to change your Kerberos password. As indicated by the brackets, the default answer is NO.

System Action: KDB@EDIT continues.

User or Operator Response: Change your password if you so desire.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2643I Random password [y] ?

Explanation: This message allows you to have Kerberos select a random password for you.

System Action: Kerberos continues.

User or Operator Response: If you want Kerberos to select a random password for you, enter `y` or `Y`.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2644I Null Key [y] ?

Explanation: Selecting the null key sets both the low and high key values to 0.

System Action: Kerberos continues.

User or Operator Response: If you do not want to set the low and high key values to 0, enter n or N. Otherwise these values are set to 0, which is the default.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2645I Principal's new key version = *version number*

Explanation: This is the new master key version number for the newly created (or edited) Kerberos user data set.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2646I Expiration date (enter yyyy-mm-dd) [*date*] ?

Explanation: This is the expiration date for this Kerberos principle.

System Action: Kerberos continues.

User or Operator Response: Change the expiration date as necessary.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2647I Date Invalid

Explanation: The format of the date that you entered does not match that shown in message EZB2646I.

System Action: Kerberos continues.

User or Operator Response: Reenter the date, making sure to use the indicated format (yyyy-mm-dd).

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2648I Max ticket lifetime (*5 minutes) *lifetime* **?**

Explanation: This message prompts you to set the maximum lifetime for tickets granted to this principal. The lifetime is set in increments of 5 minutes, up to a maximum of 255 increments (1275 minutes)

System Action: Kerberos continues.

User or Operator Response: Set the ticket lifetime as you so desire by entering a number between 1 and 255.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2649I Invalid, choose 0-255

Explanation: If the value you enter for the maximum ticket lifetime is not between 0 and 255, this message appears. It indicates that the ticket lifetime value is not within these limits.

System Action: Kerberos continues.

User or Operator Response: Enter a valid number for the ticket lifetime.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2650I Attributes [*number*]

Explanation: Attributes are numbers used to store the user's ticket information.

System Action: Kerberos continues.

User or Operator Response: Enter a value for attributes if you so desire.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2651I Invalid, choose 0-65535

Explanation: The value you entered for the attributes field was incorrect. You must choose a value between 0 and 65 535.

System Action: Kerberos continues.

User or Operator Response: Enter a valid value for attributes, as indicated.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2652I Attributes [*number*]

Explanation: Message EZB2651I is always followed by this message, which prompts you for a correct value for the attributes field.

System Action: Kerberos continues.

User or Operator Response: Enter a valid value for attributes, as indicated.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2653I Error updating Kerberos database

Explanation: The TSO UID does not have access to the Kerberos data base.

System Action: Kerberos continues.

User or Operator Response: Issue the KDB@EDIT command from the ADM@SRV user ID and repeat the steps to register the user or service. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2654I Edit O.K.

Explanation: The initial values or changes you entered in the Kerberos database have been performed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2655I Unchanged

Explanation: If an error occurs in updating the Kerberos database, this message appears. It usually appears directly after message EZB2653I.

System Action: Kerberos continues.

User or Operator Response: Reissue the KDB@EDIT command and repeat the steps in registering the user or service. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2656I There were more tuples found than there were space for

Explanation: More fields of data were entered for the principal name than the Kerberos database can accommodate.

System Action: Kerberos exits.

User or Operator Response: Reissue the KDB@EDIT command and repeat the steps in registering the user or service. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: change_principal()

EZB2657E Signal caught, sig = *signal number* **code =** *code number* **exiting**

Explanation: The signal for the Kerberos server to exit has been received. The signal number and return code are provided.

System Action: Kerberos exits.

User or Operator Response: Reenter the KDB@EDIT command.

System Programmer Response: None.

Source Data Set: KDB@EDIT

Procedure Name: sig_exit

EZB2658I Usage: *program name* **[-n]**

Explanation: If you enter an incorrect parameter to the KDB@EDIT command, this message shows the correct syntax for the command.

System Action: KDB@EDIT exits.

User or Operator Response: Reissue the KDB@EDIT command, making sure to use the correct syntax.

System Programmer Response: Assist the user as necessary.

Source Data Set: KDB@EDIT

Procedure Name: Usage()

EZB2660I Usage:*string -f -q*

Explanation: The principal name and ticket terminating procedure parameters are displayed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDESTROY

Procedure Name: usage

EZB2661I Tickets destroyed.

Explanation: The active Kerberos tickets were destroyed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDESTROY

Procedure Name: main

EZB2662I No tickets to destroy.

Explanation: The KDESTROY program did not detect any active tickets.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDESTROY

Procedure Name: main

EZB2663I Tickets NOT destroyed.

Explanation: This message is issued upon completion of the KDESTROY program. No tickets were destroyed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDESTROY

Procedure Name: main

EZB2664I No...

Explanation: This message is displayed with message EZB2663I to indicate that there are no queued Kerberos tickets that have not been destroyed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDESTROY

Procedure Name: main

EZB2665I Usage: *progname* **[-s] [-m] [-n]** *string string*

Explanation: This message indicates the name of the program using the Kerberos server. It also shows the syntax and parameters of the MVSKERB command which is used to start the Kerberos server. For more information about the MVSKERB command, see *OS/390 IBM Communications Server: IP Configuration Reference*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: usage() -->

EZB2666I pause_int must be between 5 and 3600 seconds.

Explanation: The value specified for the pause interval option during Kerberos invocation was not within the valid range of 5 to 3600 seconds.

System Action: Kerberos initialization halts.

User or Operator Response: Restart Kerberos, specifying a valid pause interval. For more information, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KERBERON

Procedure Name: main

EZB2667E max_age must be between one hour and three days, in seconds.

Explanation: The value entered for the max_age parameter in the configuration data set was not valid. The max_age parameter determines the maximum allowable age for the Kerberos database.

System Action: Kerberos halts.

User or Operator Response: Correct the max_age parameter in the configuration data set to be between 1 hour and 3 days, in seconds.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2668E Could not set alternate database name

Explanation: The Kerberos server is unable to access the alternate database specified in the configuration data set.

System Action: Kerberos exits.

User or Operator Response: Make sure that the alternate database specified in the configuration data set is in storage accessible to the Kerberos server. Correct the alternate data base name in the configuration data set if necessary. For more information about configuring the Kerberos server, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2669I Kerberos server starting

Explanation: The Kerberos server is starting.

System Action: The Kerberos server starts.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBERON

Procedure Name: main

EZB2670I Maximum database age: *decimal* **seconds**

Explanation: This message displayes the max_age value specified by the MVSKERB command.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2671I Sleep forever on error

Explanation: The system does not sleep (wait) for a given interval if the Kerberos server detects an error. The erroneous process is suspended forever and the server continues with other processes.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2672I Master key will be entered manually

Explanation: The master key, which is used by Kerberos as a base for coding and decoding tickets granting access to services controlled by Kerbers, will be read from a data set or entered by an operator rather than generated by Kerberos.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBERON

Procedure Name: main

EZB2673I Log file is *data set*

Explanation: This is the name of the Kerberos log file. It must be a fully qualified data set name. The default data set is ADM@SRV.KERBEROS.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2674E *progname*: **gethostname error**

Explanation: The gethostname call, which gets the name of the host processor on which the remote program is running, failed.

System Action: The socket call to the remote Kerberos server abends. Kerberos exits.

User or Operator Response: None.

System Programmer Response: Use the error message that accompanies this one to determine the exact nature of the gethostname error. For more information about socket call error messages, refer to *OS/390 IBM Communications Server: IP Application Programming Interface Guide*. Correct the socket call as necessary and reinitialize the calling application.

Source Data Set: KERBEROS

Procedure Name: main

EZB2675E *progname*: **udp/kerberos unknown service**

Explanation: No UDP transport service could be created for the remote server. The socket() call failed.

System Action: The socket on the remote Kerberos server is not bound. Kerberos exits.

User or Operator Response: None.

System Programmer Response: Make sure that the UDP protocol is supported for the socket type and domain specified in the socket() call. For more information about socket calls, refer to *OS/390 IBM Communications Server: IP Application Programming Interface Guide*. Reinitialize the calling application.

Source Data Set: KERBEROS

Procedure Name: main

EZB2676E Can't open socket

Explanation: Either the address domain requested or the socket type requested in the socket() call is incorrect.

System Action: The socket call to the remote Kerberos server fails. Kerberos exits.

User or Operator Response: None.

System Programmer Response: Make sure that the address domain specified in the socket() call is AF_INET. Make sure that the socket type specified is SOCK_DGRAM. For more information about socket calls, refer to *OS/390 IBM Communications Server: IP Application Programming Interface Guide*. Reinitialize the calling application.

Source Data Set: KERBEROS

Procedure Name: main

EZB2677E *progname*: **setsockopt (SO_REUSEADDR)**

Explanation: Either the option level for the setsockopt() call is not SOL_SOCKET or the SO_REUSEADDR socket option is not specified in the call. SO_REUSEADDR allows local addresses that are already in use to be bound. For more information about socket options, refer to *OS/390 IBM Communications Server: IP Application Programming Interface Guide*.

System Action: The socket call to the remote Kerberos server abends. Kerberos exits.

User or Operator Response: None.

System Programmer Response: Correct the setsockopt() call for the remote server. For more information about socket calls, refer to *OS/390 IBM Communications Server: IP Application Programming Interface Guide*. Reinitialize the calling application.

Source Data Set: KERBEROS

Procedure Name: main

EZB2678E *progname*: **Can't bind socket**

Explanation: The bind() cannot be accomplished. The pointer to the sockaddr structure that contains the name to be bound to the socket descriptor returned by the previous socket() call is nil.

System Action: The socket call for the remote Kerberos server abends. Kerberos exits.

User or Operator Response: None.

System Programmer Response: Make sure that the name parameter of the bind() indicates the pointer to the address structure containing the name that is to be bound to s, which is the socket descriptor returned by a previous socket() call. Reinitialize the calling application.

Source Data Set: KERBEROS

Procedure Name: main

EZB2679I Kerberos db and cache init failed = *number* **...exiting**

Explanation: Kerberos was unable to initialize the Kerberos database or any cached Kerberos information. Without access to the database or to cached information, Kerberos cannot issue or validate tickets granting access to Kerberos controlled services.

System Action: Kerberos halts.

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

System Programmer Response: Make sure that the Kerberos database and any cached information are in storage accessible to the Kerberos server, and restart the Kerberos server.

Source Data Set: KERBERON

Procedure Name: main

EZB2680I Current Kerberos master key version is *decimal*

Explanation: This is the number of the current master key version for the Kerberos server.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2681I Local realm: *realm*

Explanation: The realm identifies each independent Kerberos site. The principal name and instance are qualified by the realm to which they belong. The realm is usually the domain name.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2682E Ticket granting ticket service unknown

Explanation: The Kerberos Authentication Server (KAS) cannot find the address of the ticket granting server (service name) in the Kerberos database.

System Action: The session ticket is not generated. Kerberos exits.

User or Operator Response: Contact the system programmer.

System Programmer Response: Make sure that the address for the ticket-granting server is included in the KRB.CONF data set. For more information, see *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: KERBEROS

Procedure Name: main

EZB2683E Kerberos server terminating. Exit code=*error number*

Explanation: The Kerberos Authentication Server is terminating. The error code indicates the reason for termination. Valid error numbers are:

1 Parameter error. Refer to EZB2665I for details.

2 Could not set alternate database name. Refer to EZB2668E for details.

3 Gethostname call failed. Refer to EZB2674E for details.

4 The socket() call failed. Refer to EZB2675E for details.

5 The socket could not be opened. Refer to EZB2676E for details.

6 The socket could not be bound. Refer to EZB2678E for details.

7 Kerberos was unable to initialize the Kerberos database or any cached Kerberos information. Refer to EZB2679I for details.

8 Kerberos couldn't get the master key. Check that the database was correctly built via KDB@INIT and the key was saved via KSTASH.

9 Kerberos couldn't verify the master key. Check that the data base was correctly build via KDB@INIT and the key was saved via KSTASH.

10 Refer to EZB2682E for details.

99 An error has occurred on recvfrom. Message EZB2683E will be repeated, and the errno from recvfrom will be listed as the Exit code. Contact the system programmer.

System Action: Kerberos is terminated.

User or Operator Response: Correct the problem as indicated by the error above. Check the system log for any additional error messages.

System Programmer Response: None.

Source Data Set: KERBEROS

Procedure Name: main

EZB2685I **Ticket file:** *key_data_set*

Explanation: The Kerberos ticket data set is displayed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: main

EZB2686E *progname*: *errormsg*

Explanation: Kerberos displays the program name and an error message due to an error being detected.

System Action: KLIST continues. Kerberos continues.

User or Operator Response: Use the error message issued in this message to determine and correct the error.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: main

EZB2687I *prog_name*: **can't find realm of ticket file:** *key_data_set*

Explanation: The stated program cannot find the realm of the ticket file.

System Action: KLIST command fails. Kerberos continues.

User or Operator Response: Resubmit the KINIT command using the correct realm syntax. Resubmit the KLIST command. If the error continues contact the system programmer. For more information about the KINIT and KLIST commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Make sure the realm is defined in the KRB.CONF data set. For more information about the KRB.CONF data set see the *OS/390 IBM Communications Server: IP User's Guide*.

Source Data Set: KLIST

Procedure Name: main

EZB2688I **Principal:** *principal@address*

Explanation: This message is issued when the KLIST program is invoked.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: main

EZB2689I *issued expired principal*

Explanation: This message is issued when the KLIST command is successfully invoked.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: main

EZB2690I *issue_ date lifetime*

Explanation: The Kerberos KLIST program lists the issue date and the lifetime of the ticket as an unsigned character.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: main

EZB2692I **No tickets in file.**

Explanation: Kerberos issues this message if the KLIST program was unable to detect any service tickets while scanning the userid.TMP.TKTO data set.

System Action: KLIST command fails. Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: main

EZB2693I **Usage:** *data_set_name* **[-sl-t] [-file filename] [srvtab]**

Explanation: The current Kerberos program in use is displayed along with the program name, command parameters, and the user_id.ETC.SRVTAB command which lists the contents of the key data set.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: usage

EZB2694I Server key file: *key_data_name*

Explanation: The Kerberos server key data name is displayed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: usage

EZB2695I *service instance realm keyversion*

Explanation: The Kerberos server header for the key data set is displayed.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: usage

EZB2697E key file truncated

Explanation: Kerberos was unable to read the key data set, so the operation has been terminated.

System Action: Kerberos exits.

User or Operator Response: Check the syntax of the service request and resubmit the KLIST command using the correct syntax. For more information about the KLIST command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: usage

EZB2698I *service instance realm keyversion*

Explanation: The server displays the service, instance, realm, and key version information after converting it from ASCII into EBCDIC.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KLIST

Procedure Name: usage

EZB2700I Kerberos error: *error*

Explanation: Kerberos has encountered an unexpected error. The type of error is detailed in the *error* portion of this message. The possible errors are listed alphabetically in the table below. The function returning the error is indicated in parentheses at the end of each message.

Message	Explanation
OK	Kerberos reported an error when none existed. Kerberos continues, no intervention is required.

Password incorrect

An incorrect password has been submitted to the Kerberos Authentication Server (KAS) in a request for access to a protected ticket-granting server (TGS). A ticket is not granted. Kerberos continues. The user should resubmit the request, making sure that the password is correctly spelled and the request is in the proper format. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Warning: Not ALL tickets returned

All of the tickets issued by the Kerberos Authentication Server (KAS) and ticket-granting server (TGS) have not been returned by the clients to whom they were issued. Tickets are reusable and have a typical lifetime of 8 hours, so a missing ticket does represent a security risk. The system programmer should use the Kerberos Database Manager (KDBM) to determine which of the tickets have not been returned and to what clients they were issued, then use trace messages to find those tickets. If the tickets cannot be found, the tickets should be removed from the Kerberos database to prevent unauthorized use.

Don't have ticket granting ticket (get_ad_tkt)

The Kerberos ticket-granting server (TGS) does not have the proper authority to grant a ticket to the client, allowing the client to use the TGS. This indicates that the TGS is configured incorrectly. Kerberos halts and the connection between the client and the TGS is closed. TCPIP continues. The system programmer should stop the TGS and take it off the network, then check the configuration to determine why the TGS is unable to grant a ticket to the client. For more information on configuring the Kerberos TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Generic error (get_intkt)

Kerberos has encountered an unrecognized failure in the ticket granting process. Kerberos halts and the connection betweeen the client and the servers is closed. The system programmer should stop the Kerberos Authentication Server (KAS) and the ticket-granting server (TGS), and check to make sure that they are properly configured. For more information about configuring the KAS and TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Protocol error (get_intkt)

The protocols being used by the client, the Kerberos Authentication Server (KAS), and the ticket-granting server (TGS) are not compatable. Kerberos halts. The system programmer should check the client, the KAS, and the TGS to make sure that they are using compatable protocols and replace any incompatable version of Kerberos being used to a compatable one.

Current password incorrect (get_pw_tkt)

The password submitted to the Kerberos Authentication Server (KAS) in a request for access to a ticket-granting server (TGS) was incorrect. Kerberos halts and no connection is opened between the client and the TGS. The user should resubmit the request, making sure that the password is spelled correctly and submitted in the correct form. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Current password is NULL (get_pw_tkt)

The current password is a null, or empty password. A null password can be used for general access to non-protected areas of a network. The Kerberos Authentication Server (KAS) will not grant access to a protected server to a user with a null password. The user should contact the system administrator to obtain a password for the protected server.

Error returned by KDC (gt_pw_tkt)

The key distribution center (KDC) which grants keys to encrypt and decrypt coded Kerberos tickets to the client and the ticket-granting server (TGS) encountered an error. Kerberos halts. The system programmer should check the configuration of the KDC to make sure that it is correct. For more information on configuring the KDC, see the *OS/390 IBM Communications Server: IP Configuration Reference*

Null ticket returned by KDC (gt_pw_tkt)

The key distribution center (KDC), which grants keys to encode and decode tickets to the client and the ticket-granting server (TGS), returned an empty ticket. Kerberos will not allow access with an empty ticket. The client is not granted access to the TGS. The system programmer should check the KDC to make sure that it is properly configured. For more information about configuring the KDC, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Protocol error (gt_pw_tkt)

The versions of Kerberos being used by the client, the Kerberos Authentication Server (KAS), and the

ticket-granting server (TGS) are not compatible. Kerberos halts. The system programmer should check the client, the KAS, and the TGS to determine which is running the incompatable protocol and replace the incompatable protocol with a compatable one.

Authentication expired (kerberos)
A connection to a server cannot be established because the authentication granted by the Kerberos Authentication Server (KAS) has expired. A new authentication is required each time a connection is established to a server. The connection is not established. The user should request a new authentication from the KAS using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Bad byte order (kerberos)
A ticket has been received from either the client, the Kerberos Authentication Server (KAS), or the ticket-granting server (TGS) that is not correctly formatted. Kerberos halts. The system programmer should check each of these devices to determine which is sending incorrectly formatted tickets, and why. For information on configuring Kerberos devices, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Permission Denied (kerberos)
Either the Kerberos Authentication Server (KAS) or the ticket-granting server (TGS) has refused to open a connection between the client and a Kerberos protected server because the client is not authorized to use the server. The user should see the system administrator for access to this server.

Principal expired (kerberos)
The Kerberos Authentication Server (KAS) has refused to authorize a connection between a client and a ticket-granting server (TGS) because the principal authorization for the client's Kerberos session has expired. The client should begin another Kerberos session using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Principal has null key (kerberos)
The principal service provided by the Kerberos Authentication Server (KAS) has a null, or empty, key. The keys are used to provide encoding and decoding of tickets for the KAS and the ticket- granting server (TGS). The key being used by the principal service contains no data, so tickets cannot be coded or decoded. The system programmer should check the KAS to determine why it is issuing a null key. For more information about the KAS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Principal: Incorrect master key version (kerberos)
The master key being used by the principal service for Kerberos is not compatable with the master key version specified in the configuration of the Kerberos Authentication Server (KAS). Without the correct master key version, no Kerberos service can be established. The system programmer should check the principal service to determine why it is using an incorrect master key version and correct the configuration to avoid this message. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Principal not unique (kerberos)
The principal identifier submitted to the Kerberos Authentication server (KAS) was not unique. The principal name must be unique for each client and service assigned by the Kerberos Manager. The user should resubmit the request, making sure that the principal name is spelled correctly and in the correct format. For more information about the principal name, see the *OS/390 IBM Communications Server: IP User's Guide*.

Principal unknown (kerberos)
The principal name submitted to the Kerberos Authentication Server (KAS) was not recognized by the KAS. The KAS will not authorize a connection to a ticket-granting server (TGS). The user should contact the system administrator to get access to the TGS.

Service expired (kerberos)
The service authorized by the Kerberos Authenication Server (KAS) has expired. No connection to the host is established. The user should restart Kerberos service using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Service: Incorrect master key version (kerberos)
A Kerberos service defined in the configuration data set for the Kerberos Authentication Server (KAS) has an incorrect master key version. Without the proper master key version, no Kerberos service can be established. They system programmer should check the configuration of the KAS and correct the master key version specified for the indicated service. For more information about configuring the KAS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Generic kerberos error (kfailure)
Kerberos has encountered an unexpected error while opening service for a client. Kerberos service halts. The system programmer should check the configuration data sets for the Kerberos Authentication Server (KAS), the ticket-granting server (TGS), and the client, for accuracy and compatability, and generate a trace of the Kerberos service if necessary to determine the cause of the error. For information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*. If more assistance is required, contact the IBM Software Support Center.

Can't find ticket (krb_get_cred)
The Kerberos Authentication Server (KAS) is unable to find a ticket for the service name, instance, and realm specified by the client. No service can be established without a ticket. The user should make sure that the service name, instance, and realm were specified properly. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*. If the error persists, the system programmer should check the configuration of the KAS to make sure that credential information is supplied for the indicated service name, instance, and realm. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Can't read ticket file (krb_get_cred)
The Kerberos Authentication Server (KAS) was unable to read the ticket file. The ticket file contains ticket credentials for specified service names, incidences, and realms. No Kerberos service can be established without a proper ticket. The system programmer should make sure that the ticket file is properly configured and in storage accessible to the KAS. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Ticket granting ticket expired (krb_mk_req)
The Kerberos ticket-granting server (TGS) is no longer authorized to grant tickets to clients requesting service because the ticket-granting ticket has expired. The user should begin a new Kerberos session using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Can't decode authenticator (krb_rd_req)
The Kerberos ticket-granting server (TGS) is unable to decode the authenticator received from a client, who received it from the Kerberos Authentication Server (KAS). This indicates that either the KAS or the TGS is configured incorrectly. No service can be established until the authenticator has been decoded. The system programmer should check the KAS and TGS to make sure that they are properly configured. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Illegal message type (krb_rd_req)
The Kerberos Authentication Server (KAS) received a message of an incorrect type from a client. This indicates that the client is incorrectly configured. No service can be established until the client, the KAS, and the ticket-granting server (TGS) are all using the same configuration. The system programmer should check the client and make any necessary corrections to the configuration. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Incorrect network address (krb_rd_req)
The Kerberos Authentication Server (KAS) has received a message from a client that gives a nonexistent network address. Kerberos is unable to verify the identity of the client because it does not recognize the network address. No service is established. The system programmer should check the client to make sure that it is specifying the proper network address. The system programmer should also check the KAS to make sure that all network addresses requiring Kerberos service are listed in the Kerberos Database. For more information about configuring Kerberos and the Kerberos Database, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Message duplicate or out of order (krb_rd_req)
The Kerberos Authentication Server (DAS) has received a message from a client that is either a duplicate message or was sent out of sequence. This indicates that the client is configured incorrectly. No service is established. The system programmer should check the configuration of the client and correct any errors. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Message integrity error (krb_rd_req)
The Kerberos Authentication Server (KAS) received a message from a client that was not intact. The KAS will

not grant authorization for a client sending corrupted packets. No service will be established. The system programmer should check the client's connections to the network to determine why the message was not received intact.

Protocol version mismatch (krb_rd_req)
The Kerberos Authentication Server (KAS), the ticket-granting server (TGS), and the client are not all using the same Kerberos protocol. Authentication cannot be established unless these hosts are all using the same Kerberos protocol. No service is established. The system programmer should check each of these devices to determine which one caused the protocol mismatch and correct the protocol specification in the appropriate configuration. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Repeat request (krb_rd_req)
The Kerberos Authentication Server (KAS) received a duplicate request from a client. This indicates that the client is not receiving messages sent by the KAS. No service is established. The system programmer should check the KAS and its connections to the network. If the connections are good, the system programmer should check the client to determine why it is sending duplicate requests. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Request inconsistent (krb_rd_req)
The principal name, instance, and realm submitted in a request for Kerberos service from a client were conflicting, or did not go together. The Kerberos Authentication Server (KAS) will not grant service based on this information. The operator should check the spelling and syntax of the principal name, instance, and realm, and resubmit them. If the error persists, the system programmer should check the Kerberos Database to make sure that all valid principal names, instances, and realms, are listed with the Kerberos Database Manager (KDBM) For more information on Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*, for more information on the KDBM, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Ticket expired (krb_rd_req)
A client cannot establish service from a Kerberos ticket-granting server because the client's ticket has expired and is no longer valid. The user should begin a new Kerberos session using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Ticket for wrong server (krb_rd_req)
A client cannot establish service from a Kerberos ticket-granting server because the ticket held by the client is for a different server. No service is established. The user should open a connection to the Kerberos Authentication Server (KAS) using the KINIT command and request a ticket for the proper server. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Ticket issue date too far in the future (krb_rd_req)
A client cannot establish service with a Kerberos ticket-granting server (TGS) because the request for a ticket specified that the ticket is to be issued farther in the future than Kerberos allows. No service is established. The client should resubmit the request, specifying a time less than 24 hours in the future. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Time is out of bounds (krb_rd_req)
A client is unable to establish service from a ticket-granting server (TGS) because either the difference in the client's clock and the server's clock is too great or the difference in the Kerberos clock and the server clock is too great. No service is established. The system programmer should check the Kerberos Authentication Server (KAS), the TGS, and the client and reset the asynchronous clock.

Unauthorized request (krb_rd_req)
A client is unable to establish service from a ticket-granting server (TGS) because the client is not authorized to receive the requested service. No service is established. The client should contact the system administrator for authorization to use the service.

Can't send request (send_to_kdc)
A client is unable to send a message to the Kerberos Authentication Server (KAS) for a key to be used in encoding and decoding authentications. No service is established. The system programmer should check the client and the KAS to determine why the request cannot be sent. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Retry count exceeded (send_to_kdc)
A client has exceeded the maximum number of attempts to retry sending a message to the Kerberos Authentication Server (KAS). This indicates that KAS is not receiving or not acknowledging the message. No

service is established. The system programmer should check the client and the KAS to determine if the message is not being sent or not being acknowledged and respond accordingly.

Bad ticket file format (tf_util)
The ticket-granting server (TGS) is unable to read the ticket file because the ticket file is formatted incorrectly. No service is established. The system programmer should check the ticket file and correct the format. For information about formating the TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Can't access ticket file (ft_util)
The ticket-granting server (TGS) cannot access the ticket file, which contains information about valid tickets to be issued by the TGS. No service is established. The system programmer should make sure that the ticket file is loaded and in storage accessible to the TGS. For more information about the configuring the TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Can't lock ticket file; try later (tf_util)
The function tf_init, which opens and locks the ticket file, was unable to lock the ticket file. Because the ticket file is shared by all ticket-granting servers (TGSs) in a realm, the file must be locked while it is open to prevent disparities in the ticket file. No service can be established with a TGS until the ticket file can be locked. When writing to the ticket file, a TGS locks the file exclusively for its own use. When reading a file, a TGS locks the file to prevent data from being written to the file. The function tf_util could be unable to lock the file because another TGS has already placed the file under an exclusive lock. The user should wait a reasonable period and attempt to establish service at a later time. If the error persists, the system programmer should check the TGS to determine why it is unable to lock the ticket file.

Read ticket file before tf_init (tf_util)
The ticket-granting server (TGS) read the ticket file, which contains information used to generate tickets for clients, before the function tf_init was called to open and lock the file. Unless the file is locked, the information in the file cannot be verified as valid. The TGS will not grant a ticket based on unverifiable data. The user should end the Kerberos session and establish another using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP Configuration Reference*. If the error persists, the system programmer should check the TGS to make sure that it is properly configured to read the ticket file using tf_init. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*. For more information about tf_init, see the *OS/390 IBM Communications Server: IP Programmer's Reference*.

System Action: See the table above.

User or Operator Response: See the table above.

System Programmer Response: See the table above.

Source Data Set: KPASSWD

Procedure Name: main

EZB2701E Bad name: *string*

Explanation: Kerberos displays this message when an incorrect user name is entered.

System Action: Kerberos continues.

User or Operator Response: Reenter the name with the correct syntax using the KPASSWD command. For more information about the KPASSWD command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2702E Bad instance: *string*

Explanation: Kerberos issues this message when the instance submitted using the KPASSWD command did not match the instance name in the key data set.

System Action: Kerberos continues.

User or Operator Response: Correct the syntax of the instance and resubmit the KPASSWD command. For more information about the KPASSWD command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2703E Bad realm: *string*

Explanation: Kerberos displays this message when an incorrect realm is detected.

System Action: Kerberos continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: The realm must be specified in the first line of the *user_id*.KRB.CONF data set when setting up the Kerberos system. The same realm name must be initialized in the KDB@INIT command. Make sure the two coincide. For more information about the *user_id*.KRB.CONF data set and the KDB@INIT command, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: KPASSWD

Procedure Name: main

EZB2704E Password NOT changed.

Explanation: The current Kerberos password was not altered.

System Action: Kerberos continues.

User or Operator Response: Resubmit the KPASSWD command to change the current password. The previous password submitted under this command was not different from the initial password. For more information about the KPASSWD command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2705I Password changed.

Explanation: Kerberos issues this message as a result of a successful password change.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2706E Error reading old password.

Explanation: Kerberos was unsuccessful in reading the current password while attempting to change that password.

System Action: KPASSWD fails. Kerberos continues.

User or Operator Response: Check the current password and resubmit the KPASSWD command with the correct syntax. For more information about the KPASSWD command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2707E Incorrect old password.

Explanation: Kerberos displays this message when the old password is incorrect.

System Action: Kerberos continues.

User or Operator Response: Check the old password and resubmit the KPASSWD command using the correct syntax. For more information about the KPASSWD command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2708I Kerberos error: *error_message*

Explanation: A Kerberos error occurred while attempting to change the current password. The *error_message* portion of this message gives further information about the error. The possible values of *error_message* are displayed the table below in alphabetic order.

Message
Explanation

OK Kerberos reported an error when none existed. Kerberos continues, no intervention is required.

Password incorrect
An incorrect password has been submitted to the Kerberos Authentication Server (KAS) in a request for access to a protected ticket-granting server (TGS). A ticket is not granted. Kerberos continues. The user should resubmit the request, making sure that the password is correctly spelled and the request is in the proper format. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Warning: Not ALL tickets returned
All of the tickets issued by the Kerberos Authentication Server (KAS) and ticket-granting server (TGS) have not been returned by the clients to whom they were issued. Tickets are reusable and have a typical lifetime of 8 hours, so a missing ticket does represent a security risk. The system programmer should use the Kerberos Database Manager (KDBM) to determine which of the tickets have not been returned and to what clients they were issued, then use trace messages to find those tickets. If the tickets cannot be found, the tickets should be removed from the Kerberos database to prevent unauthorized use.

Don't have ticket granting ticket (get_ad_tkt)
The Kerberos ticket-granting server (TGS) does not have the proper authority to grant a ticket to the client, allowing the client to use the TGS. This indicates that the TGS is configured incorrectly. Kerberos halts and the connection between the client and the TGS is closed. TCPIP continues. The system programmer should stop the TGS and take it off the network, then check the configuration to determine why the TGS is unable to grant a ticket to the client. For more information on configuring the Kerberos TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Generic error (get_intkt)
Kerberos has encountered an unrecognized failure in the ticket granting process. Kerberos halts and the connection betweeen the client and the servers is closed. The system programmer should stop the Kerberos Authentication Server (KAS) and the ticket-granting server (TGS), and check to make sure that they are properly configured. For more information about configuring the KAS and TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Protocol error (get_intkt)
The protocols being used by the client, the Kerberos Authentication Server (KAS), and the ticket-granting server (TGS) are not compatable. Kerberos halts. The system programmer should check the client, the KAS, and the TGS to make sure that they are using compatable protocols and replace any incompatable version of Kerberos being used to a compatable one.

Current password incorrect (get_pw_tkt)
The password submitted to the Kerberos Authentication Server (KAS) in a request for access to a ticket-granting server (TGS) was incorrect. Kerberos halts and no connection is opened between the client

and the TGS. The user should resubmit the request, making sure that the password is spelled correctly and submitted in the correct form. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Current password is NULL (get_pw_tkt)
The current password is a null, or empty password. A null password can be used for general access to non-protected areas of a network. The Kerberos Authentication Server (KAS) will not grant access to a protected server to a user with a null password. The user should contact the system administrator to obtain a password for the protected server.

Error returned by KDC (gt_pw_tkt)
The key distribution center (KDC) which grants keys to encrypt and decrypt coded Kerberos tickets to the client and the ticket-granting server (TGS) encountered an error. Kerberos halts. The system programmer should check the configuration of the KDC to make sure that it is correct. For more information on configuring the KDC, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Null ticket returned by KDC (gt_pw_tkt)
The key distribution center (KDC), which grants keys to encode and decode tickets to the client and the ticket-granting server (TGS), returned an empty ticket. Kerberos will not allow access with an empty ticket. The client is not granted access to the TGS. The system programmer should check the KDC to make sure that it is properly configured. For more information about configuring the KDC, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Protocol error (gt_pw_tkt)
The versions of Kerberos being used by the client, the Kerberos Authentication Server (KAS), and the ticket-granting server (TGS) are not compatible. Kerberos halts. The system programmer should check the client, the KAS, and the TGS to determine which is running the incompatable protocol and replace the incompatable protocol with a compatable one.

Authentication expired (kerberos)
A connection to a server cannot be established because the authentication granted by the Kerberos Authentication Server (KAS) has expired. A new authentication is required each time a connection is established to a server. The connection is not established. The user should request a new authentication from the KAS using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Bad byte order (kerberos)
A ticket has been received from either the client, the Kerberos Authentication Server (KAS), or the ticket-granting server (TGS) that is not correctly formatted. Kerberos halts. The system programmer should check each of these devices to determine which is sending incorrectly formatted tickets, and why. For information on configuring Kerberos devices, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Permission Denied (kerberos)
Either the Kerberos Authentication Server (KAS) or the ticket-granting server (TGS) has refused to open a connection between the client and a Kerberos protected server because the client is not authorized to use the server. The user should see the system administrator for access to this server.

Principal expired (kerberos)
The Kerberos Authentication Server (KAS) has refused to authorize a connection between a client and a ticket-granting server (TGS) because the principal authorization for the client's Kerberos session has expired. The client should begin another Kerberos session using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Principal has null key (kerberos)
The principal service provided by the Kerberos Authentication Server (KAS) has a null, or empty, key. The keys are used to provide encoding and decoding of tickets for the KAS and the ticket- granting server (TGS). The key being used by the principal service contains no data, so tickets cannot be coded or decoded. The system programmer should check the KAS to determine why it is issuing a null key. For more information about the KAS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Principal: Incorrect master key version (kerberos)
The master key being used by the principal service for Kerberos is not compatable with the master key version specified in the configuration of the Kerberos Authentication Server (KAS). Without the correct master key version, no Kerberos service can be established. The system programmer should check the principal

service to determine why it is using an incorrect master key version and correct the configuration to avoid this message. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Principal not unique (kerberos)
The principal identifier submitted to the Kerberos Authentication server (KAS) was not unique. The principal name must be unique for each client and service assigned by the Kerberos Manager. The user should resubmit the request, making sure that the principal name is spelled correctly and in the correct format. For more information about the principal name, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Principal unknown (kerberos)
The principal name submitted to the Kerberos Authentication Server (KAS) was not recognized by the KAS. The KAS will not authorize a connection to a ticket-granting server (TGS). The user should contact the system administrator to get access to the TGS.

Service expired (kerberos)
The service authorized by the Kerberos Authenication Server (KAS) has expired. No connection to the host is established. The user should restart Kerberos service using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Service: Incorrect master key version (kerberos)
A Kerberos service defined in the configuration data set for the Kerberos Authentication Server (KAS) has an incorrect master key version. Without the proper master key version, no Kerberos service can be established. They system programmer should check the configuration of the KAS and correct the master key version specified for the indicated service. For more information about configuring the KAS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Generic kerberos error (kfailure)
Kerberos has encountered an unexpected error while opening service for a client. Kerberos service halts. The system programmer should check the configuration data sets for the Kerberos Authentication Server (KAS), the ticket-granting server (TGS), and the client, for accuracy and compatability, and generate a trace of the Kerberos service if necessary to determine the cause of the error. For information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*. If more assistance is required, contact the IBM Software Support Center.

Can't find ticket (krb_get_cred)
The Kerberos Authentication Server (KAS) is unable to find a ticket for the service name, instance, and realm specified by the client. No service can be established without a ticket. The user should make sure that the service name, instance, and realm were specified properly. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*. If the error persists, the system programmer should check the configuration of the KAS to make sure that credential information is supplied for the indicated service name, instance, and realm. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Can't read ticket file (krb_get_cred)
The Kerberos Authentication Server (KAS) was unable to read the ticket file. The ticket file contains ticket credentials for specified service names, incidences, and realms. No Kerberos service can be established without a proper ticket. The system programmer should make sure that the ticket file is properly configured and in storage accessible to the KAS. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Ticket granting ticket expired (krb_mk_req)
The Kerberos ticket-granting server (TGS) is no longer authorized to grant tickets to clients requesting service because the ticket-granting ticket has expired. The user should begin a new Kerberos session using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Can't decode authenticator (krb_rd_req)
The Kerberos ticket-granting server (TGS) is unable to decode the authenticator received from a client, who received it from the Kerberos Authentication Server (KAS). This indicates that either the KAS or the TGS is configured incorrectly. No service can be established until the authenticator has been decoded. The system programmer should check the KAS and TGS to make sure that they are properly configured. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Illegal message type (krb_rd_req)
The Kerberos Authentication Server (KAS) received a message of an incorrect type from a client. This indicates that the client is incorrectly configured. No service can be established until the client, the KAS, and the ticket-granting server (TGS) are all using the same configuration. The system programmer should check the client and make any necessary corrections to the configuration. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Incorrect network address (krb_rd_req)
The Kerberos Authentication Server (KAS) has received a message from a client that gives a nonexistent network address. Kerberos is unable to verify the identity of the client because it does not recognize the network address. No service is established. The system programmer should check the client to make sure that it is specifying the proper network address. The system programmer should also check the KAS to make sure that all network addresses requiring Kerberos service are listed in the Kerberos Database. For more information about configuring Kerberos and the Kerberos Database, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Message duplicate or out of order (krb_rd_req)
The Kerberos Authentication Server (DAS) has received a message from a client that is either a duplicate message or was sent out of sequence. This indicates that the client is configured incorrectly. No service is established. The system programmer should check the configuration of the client and correct any errors. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Message integrity error (krb_rd_req)
The Kerberos Authentication Server (KAS) received a message from a client that was not intact. The KAS will not grant authorization for a client sending corrupted packets. No service will be established. The system programmer should check the client's connections to the network to determine why the message was not received intact.

Protocol version mismatch (krb_rd_req)
The Kerberos Authentication Server (KAS), the ticket-granting server (TGS), and the client are not all using the same Kerberos protocol. Authentication cannot be established unless these hosts are all using the same Kerberos protocol. No service is established. The system programmer should check each of these devices to determine which one caused the protocol mismatch and correct the protocol specification in the appropriate configuration. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Repeat request (krb_rd_req)
The Kerberos Authentication Server (KAS) received a duplicate request from a client. This indicates that the client is not receiving messages sent by the KAS. No service is established. The system programmer should check the KAS and its connections to the network. If the connections are good, the system programmer should check the client to determine why it is sending duplicate requests. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Request inconsistent (krb_rd_req)
The principal name, instance, and realm submitted in a request for Kerberos service from a client were conflicting, or did not go together. The Kerberos Authentication Server (KAS) will not grant service based on this information. The operator should check the spelling and syntax of the principal name, instance, and realm, and resubmit them. If the error persists, the system programmer should check the Kerberos Database to make sure that all valid principal names, instances, and realms, are listed with the Kerberos Database Manager (KDBM) For more information on Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*, for more information on the KDBM, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Ticket expired (krb_rd_req)
A client cannot establish service from a Kerberos ticket-granting server because the client's ticket has expired and is no longer valid. The user should begin a new Kerberos session using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Ticket for wrong server (krb_rd_req)
A client cannot establish service from a Kerberos ticket-granting server because the ticket held by the client is for a different server. No service is established. The user should open a connection to the Kerberos Authentication Server (KAS) using the KINIT command and request a ticket for the proper server. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Ticket issue date too far in the future (krb_rd_req)
A client cannot establish service with a Kerberos ticket-granting server (TGS) because the request for a ticket specified that the ticket is to be issued farther in the future than Kerberos allows. No service is established. The client should resubmit the request, specifying a time less than 24 hours in the future. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP User's Guide*.

Time is out of bounds (krb_rd_req)
A client is unable to establish service from a ticket-granting server (TGS) because either the difference in the client's clock and the server's clock is too great or the difference in the Kerberos clock and the server clock is too great. No service is established. The system programmer should check the Kerberos Authentication Server (KAS), the TGS, and the client and reset the asynchronous clock.

Unauthorized request (krb_rd_req)
A client is unable to establish service from a ticket-granting server (TGS) because the client is not authorized to receive the requested service. No service is established. The client should contact the system administrator for authorization to use the service.

Can't send request (send_to_kdc)
A client is unable to send a message to the Kerberos Authentication Server (KAS) for a key to be used in encoding and decoding authentications. No service is established. The system programmer should check the client and the KAS to determine why the request cannot be sent. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Retry count exceeded (send_to_kdc)
A client has exceeded the maximum number of attempts to retry sending a message to the Kerberos Authentication Server (KAS). This indicates that KAS is not receiving or not acknowledging the message. No service is established. The system programmer should check the client and the KAS to determine if the message is not being sent or not being acknowledged and respond accordingly.

Bad ticket file format (tf_util)
The ticket-granting server (TGS) is unable to read the ticket file because the ticket file is formatted incorrectly. No service is established. The system programmer should check the ticket file and correct the format. For information about formating the TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Can't access ticket file (ft_util)
The ticket-granting server (TGS) cannot access the ticket file, which contains information about valid tickets to be issued by the TGS. No service is established. The system programmer should make sure that the ticket file is loaded and in storage accessible to the TGS. For more information about the configuring the TGS, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

Can't lock ticket file; try later (tf_util)
The function tf_init, which opens and locks the ticket file, was unable to lock the ticket file. Because the ticket file is shared by all ticket-granting servers (TGSs) in a realm, the file must be locked while it is open to prevent disparities in the ticket file. No service can be established with a TGS until the ticket file can be locked. When writing to the ticket file, a TGS locks the file exclusively for its own use. When reading a file, a TGS locks the file to prevent data from being written to the file. The function tf_util could be unable to lock the file because another TGS has already placed the file under an exclusive lock. The user should wait a reasonable period and attempt to establish service at a later time. If the error persists, the system programmer should check the TGS to determine why it is unable to lock the ticket file.

Read ticket file before tf_init (tf_util)
The ticket-granting server (TGS) read the ticket file, which contains information used to generate tickets for clients, before the function tf_init was called to open and lock the file. Unless the file is locked, the information in the file cannot be verified as valid. The TGS will not grant a ticket based on unverifiable data. The user should end the Kerberos session and establish another using the KINIT command. For more information about Kerberos commands, see the *OS/390 IBM Communications Server: IP Configuration Reference*. If the error persists, the system programmer should check the TGS to make sure that it is properly configured to read the ticket file using tf_init. For more information about configuring Kerberos, see the *OS/390 IBM Communications Server: IP Configuration Reference*. For more information about tf_init, refer to the *OS/390 IBM Communications Server: IP Programmer's Reference*.

System Action: See the table above.

User or Operator Response: See the table above.

System Programmer Response: See the table above.

Source Data Set: KPASSWD

Procedure Name: main

EZB2709E Null passwords are not allowed: try again.

Explanation: Kerberos detected a null password while attempting to create a new password.

System Action: Kerberos continues.

User or Operator Response: Resubmit the KPASSWD command. The password must contain from 1 to 8 characters. For more information on the KPASSWD command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2710I Usage: kpasswd [-h] [-n user] [-i instance] [-r realm] \[-u fullname]

Explanation: This message only occurs when the KPASSWD command cannot be implemented. The correct syntax and parameters for the KPASSWD command are displayed following the KPASSWD failure.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KPASSWD

Procedure Name: main

EZB2711E Kerberos db and cache init failed = *errorcode*

Explanation: Kerberos was unable to initialize the Kerberos database or any cached Kerberos information. Without access to the database or to cached information, Kerberos cannot issue or validate tickets granting access to Kerberos controlled services.

System Action: Kerberos halts.

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

System Programmer Response: Make sure that the Kerberos database and any cached information are in storage accessible to the Kerberos server, and restart the Kerberos server.

Source Data Set: KSTASH

Procedure Name: main

EZB2712E *application***: Couldn't read master key.**

Explanation: The indicated application was unable to read the master key used to code and decode Kerberos tickets granting access to services controlled by Kerberos.

System Action: Kerberos halts.

User or Operator Response: Make sure that the master key data set is loaded and in storage accessible to the application. Make sure that the master key data set is of the correct format. For more information, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KSTASH

Procedure Name: main

EZB2713E *application*: **Unable to open master key file**

Explanation: The indicated application was unable to open the data set containing the master key used to code and decode tickets granting access to services controlled by Kerberos.

System Action: Kerberos halts.

User or Operator Response: Make sure that the master key data set is loaded and in storage accessible to the application. Make sure that the master key data set is of the proper format. For more information, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KSTASH

Procedure Name: main

EZB2714E *application*: **Write I/O error on master key file**

Explanation: The indicated application encountered a write I/O error while processing the master key data set.

System Action: Kerberos halts.

User or Operator Response: Make sure that the application has the proper authority to write to the master key data set.

System Programmer Response: None.

Source Data Set: KSTASH

Procedure Name: main

EZB2717E \a\a\n %s - Record Number %ld

Explanation: The alert bell indicates the following number encountered an error when trying to open or close the data set or read or write in the data set.

System Action: The task exits.

User or Operator Response: Check and make sure the data set is accessible and data is currently in the data set.

System Programmer Response: Assist the user, if necessary.

Source Data Set: BPLUS1

Procedure Name: error

EZB2718E Could not create a new file

EZB2719E File already exists.

Explanation: A new data set cannot be generated because of a existing data set.

System Action: The task continues.

User or Operator Response: The existing data set can be deleted or a new data set can be created. Resubmit the request after one of the preceding procedures have been submitted.

System Programmer Response: None.

Source Data Set: BPLDBM

Procedure Name: bpl_open

EZB2720E bpl_fetch: key was not set

Explanation: The key must be in the same position in each data set, and the key data must be contiguous to read data found in the data set.

System Action: The task exits.

User or Operator Response: Make sure data resides in the data set and the key points to information within the data set.

System Programmer Response: None.

Source Data Set: BPLDBM

Procedure Name: bpl_nextkey

EZB2721E kerb_dbl_init: couldn't open *name*

Explanation: The kerb_dbl_init function encountered an error when opening the data set name specified.

System Action: The task exits.

User or Operator Response: Make sure the Kerberos data base is defined in the ADM@SRV user ID.

System Programmer Response: None.

Source Data Set: KRB@DBM

Procedure Name: kerb_dbl_init

EZB2722I *name instance* **expires** *year month day hour min* **max_life** *max_life* **=** *number* **min attr** *attributes*

Explanation: The indicated ticket will expire at the specified time. This message also displays the maximum life of the ticket, and its attributes.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PRINT@PR

Procedure Name: kerb_print_principal

EZB2723I tkey_ver *version* **k_low** *low_ticket* **k_high** *high_ticket* **akv** *number* **exists** *old_ticket*

Explanation: This message displays the master key version, low and high key numbers, server key version, and the previous ticket number for a newly issued Kerberos ticket.

System Action: Kerberos continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: PRINT@PR

Procedure Name: kerb_print_principal

EZB2725E Usage: *command* **operation file-name {database name}.**

Explanation: The *KDB@UTIL* parameter indicates the database name from where the data is being access from.

System Action: The KDB@UTIL command exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: main

EZB2726E *command string* **is an invalid operation.**

Explanation: The indicated operation cannot be performed, due to the syntax of the parameter. This message will be followed by the correct syntax that can used with the *KDB@UTIL* parameter.

System Action: The KDB@UTIL command exits.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: main

EZB2727I *command* **Valid operations are \″dump\″, \″load\″.**

Explanation: These commands can be used in conjunction with the *KDB@UTIL* command to build and format the Kerberos database.

System Action: The system waits for the user's response.

User or Operator Response: Resubmit the *KDB@UTIL* command using one of the valid operations. For more information see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: main

EZB2728E *command* **Unable to open** *data set name*

Explanation: A error was detected when trying to open the data set specified. The data set must be created in ADM@SRV before execution can start. The data base is not accessible.

System Action: TCPIP continues.

User or Operator Response: Edit the *user_id.KRB.CONF* data set to make sure the data set is properly set. If necessary, create a Kerberos data base. For more information about creating a Kerberos data base, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: main

EZB2729E **error on file** *data set name*

Explanation: The data set was not able to dump successfully.

System Action: KDB@UTIL exits.

User or Operator Response: Edit the data set and make sure the data set uses the correct syntax.

System Programmer Response: Assist the user if necessary.

Source Data Set: KDB@UTIL

Procedure Name: main

EZB2730I **Don't forget to do a kdb_util load** *name* **to reload the database!**

Explanation: After a *DUMP* has been performed a *LOAD* operation must done an order to reload the database. The indicated data set name contains the datebase data and must be used with this function.

System Action: KDB@UTIL continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: main

EZB2731E Couldn't create temp database *name*

Explanation: The temporary database encountered an error during creation.

System Action: KDB@UTIL continues.

User or Operator Response: Use the *KDB@INIT* command to create a database. Information pertaining to this command can be found in the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: load_db

EZB2732E couldn't store *name* **.** *instance***; load aborted**

Explanation: The contents of the data set name cannot be stored in the Kerberos database. This error can occur due to the inadequate storage space.

System Action: KDB@UTIL continues.

User or Operator Response: Contact the system programmer.

System Programmer Response: Use standard procedures to allocate more storage space for this application.

Source Data Set: KDB@UTIL

Procedure Name: updata_ok_file

EZB2733E kdb_util: out of memory.

Explanation: The amount of storage space is not available to accommodate the database data.

System Action: KDB@UTIL continues.

User or Operator Response: Increase the region size and reissue the KDB@UTIL command.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: updata_ok_file

EZB2734E Error creating 'ok' file, '*name***'**

Explanation: An error was encountered when generating the indicated data set name.

System Action: The procedure is discontinued until the error is corrected.

User or Operator Response: Make sure the Kerberos data set exists for the ADM@SRV user ID.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: update_ok_file

EZB2735I \n\nEnter the CURRENT master key.

Explanation: This message prompts you for the master key.

System Action: KDB@UTIL continues.

User or Operator Response: The master key must be entered and is case-sensitive.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_new_master_key

EZB2736E *command* **Couldn't get master key.**

Explanation: The master key is not defined in the Kerberos database.

System Action: TCPIP continues.

User or Operator Response: Issue the KDB@INIT command to create a master key.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_new_master_key

EZB2737I Now enter the NEW master key. Do not forget it!!

Explanation: The new master key must be entered to access the database data sets found in Kerberos. This message only occurs when a new Kerberos database is created.

System Action: KDB@UTIL continues.

User or Operator Response: Enter the master key using the correct, case-sensitive syntax. Note new master key for future use.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_new_master_key

EZB2738E Couldn't get new master key.

Explanation: The ADM@SRV cannot acknowledge the new master key used to access the database.

System Action: KDB@UTIL continues.

User or Operator Response: Make sure the master key is stored. For more information on this command see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: kdb_get_master_key

EZB2739E *command* **Couldn't get master key.**

Explanation: The master key specified cannot recognized by the Kerberos database.

System Action: KDB@UTIL exits.

User or Operator Response: Reenter the master key using case-sensitive syntax. For more information see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: kdb_get_master_key

EZB2740E verify_master_key : Kerberos error on master key lookup, *number* **found.**

Explanation: The master key version specified cannot be recognized by Kerberos. Kerberos can only support 1 master key version.

System Action: The KDB@UTIL command exits.

User or Operator Response: The master key cannot be initialized unless the master key version is updated for the database.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_old_format_db

EZB2741I Current Kerveros master key version is *version number*

Explanation: This message displays the current Kerberos master key version.

System Action: The KDB@UTIL command continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_old_format_db

EZB2742E %verify_master_key: Invalid master key does not match database.

Explanation: The master key does not match the key in the Kerberos database. For more information see section *Communicating with the Authentication Server* in the *OS/390 IBM Communications Server: IP Programmer's Reference*.

System Action: The KDB@UTIL command continues.

User or Operator Response: Reenter the master key.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_old_format_db

EZB2743I Master key verified.

Explanation: The master key verified with the Kerberos database master key.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@UTIL

Procedure Name: convert_old_format_db

EZB2745I Verifying, please re-enter

Explanation: After Kerberos prompts you for the new password, the new password must be re-entered to be verified. If the new password is invalid the following message EZB2746E will display.

System Action: The task continues.

User or Operator Response: Re-enter the new password.

System Programmer Response: None.

Source Data Set: GET@IN@T, GET@INTN

Procedure Name: push_signals

EZB2746E Mismatch - try again

Explanation: The password you have entered does not match the origin password.

System Action: The system waits for the user's response.

User or Operator Response: Check the spelling when entering the password.

System Programmer Response: None.

Source Data Set: GET@IN@T, GET@INTN

Procedure Name: push_signals

EZB2747E unknown field value

Explanation: A field value must be entered after the following prompts: NAME, INSTANCE, or REALM. The instance and realm fields can use a default null.

System Action: The task continues.

User or Operator Response: Reenter the command with the correct field. For more information on these fields see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KNAME@PA

Procedure Name: kname_parse

EZB2748E nkrb_rd_safe protocol err sizeof(u_long) != sizeof(struct in_addr)

Explanation: The size of the structures cannot be placed into the storage space provided, therefore, the structures cannot be sent. The storage space must be increased to accommodate the data.

System Action: TCPIP cannot transmit the data until the error is corrected.

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

System Programmer Response: Use the standard system procedure to increase the size of the storage space.

Source Data Set: RD@SAFE, RD@SAFEN

Procedure Name: krb_rd_safe

EZB2750E Verifying, please re-enter

Explanation: After Kerberos prompts you for the new password, the new password must be reentered to be verified. If the new password is a mismatch, message EZB2751I will follow.

System Action: The system continues processing.

User or Operator Response: Reenter the new password. For more information on the usage of the password parameter see, *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: READ@PAS

Procedure Name: push_signals

EZB2751E Mismatch - try again

Explanation: The password you have entered does not match the previous entered password.

System Action: The system waits for the users response.

User or Operator Response: Reenter the password with the correct syntax or create a new password by using the KPASSWD command. For more information about *passwd* see, *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist user if necessary.

Source Data Set: READ@PAS

Procedure Name: push_signals

EZB2755E Usage: *command*

Explanation: This message displays the correct usage of the *KDB@INIT* command. For more information on the *KDB@INIT* command see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: KDB@INIT exits.

User or Operator Response: Resubmit the command using the correct syntax.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2756E Couldn't create database

Explanation: A database cannot be create, if a data set is currently in the database. To create a new database, the existing data set must be discarded. This can be accomplish by using *KDB@DEST* parameter to discard the current data set and *KDB@INIT* parameter to create a new data set.

System Action: KDB@INIT exits.

User or Operator Response: For more information on the two parameters use to create and destroy a database see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the user if necessary.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2757I Realm name {default *default name***}:**

Explanation: This message prompts the user for the realm name. If a realm name is not indicated the default realm name is used. The default realm name is *YOUR_KRB.REALM*.

System Action: KDB@INIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2758E EOF reading realm

Explanation: An error or end-of-data condition occurred before completing the read execution. This can be caused by a new-line character not found in the realm data set or the size of the realm storage was not allocated at the beginning of the realm string.

System Action: KDB@INIT exits.

User or Operator Response: Make sure the realm data set is open for input. Also make sure a new-line character is included, otherwise characters are transmitted as read.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2759E Bad kerberos realm name *name*

Explanation: Kerberos cannot indentify the realm name specified.

System Action: KDB@INIT exits.

User or Operator Response: Reissue the KDB@INIT command.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2760I You will be prompted for the database Master Password.

EZB2761I It is important that you NOT FORGET this password.

Explanation: A prompt will follow these messages for the password to access the Kerberos database.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2762E Couldn't read master key.

Explanation: The master key was not recognized by the Kerberos database data set. Message EZB2763E will display, if the condition is incorrect.

System Action: KDB@INIT exits.

User or Operator Response: Reissue the KDB@INIT command with the proper master key.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2763E *command* **couldn't initialize database**

Explanation: KDB@INIT cannot initialize the database. The Kerberos master key must be verified before the database can be utilized.

System Action: KDB@INIT exits.

User or Operator Response: Reissue the KDB@INIT command with the proper master key.

System Programmer Response: None.

Source Data Set: KDB@INIT

Procedure Name: main

EZB2770E %s: %s

Explanation: This message states the incorrect usage of the Kerberos name specifications. The Kerberos name format consists of: principle name, instance, and realm.

System Action: TCPIP waits for the user response.

User or Operator Response: Reissue the Kerberos name using the preceding format. For more details about the name formats used for the Kerberos name structure see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2771E %s: k_gethostname failed

Explanation: The k_gethostname function, which returns the name of the host processor on which the program is running, failed.

System Action: TCPIP continues.

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

System Programmer Response: Make sure the buffer parameter does not call a address space outside of the caller's address space.

Source Data Set: KINIT

Procedure Name: main

EZB2772I Kerberos Initialization for *name*

Explanation: This message indicates the data set name that was used to initialized the Kerberos procedure.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: usage

EZB2773I Kerberos Initialization

Explanation: Kerberos has now been initialized. Immediately following this message the system will prompt for the Kerberos name.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2774I Kerberos name:

Explanation: This message prompts you for the Kerberos name.

System Action: TCPIP continues.

User or Operator Response: Enter the name of the principal, instance, and realm at the given prompt.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2775E *command* **bad Kerberos name format**

Explanation: A invalid Kerberos name format was encountered when using the KINIT command.

System Action: KINIT exits.

User or Operator Response: Re-enter the name using a valid Kerberos name. Ensure the Kerberos name does not exceed 39 characters.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2776I Kerberos instance:

Explanation: This message prompts the user for the Kerberos instance.

System Action: KINIT continues.

User or Operator Response: Enter the Kerberos instance at the appropriate prompt. The *-i* parameter can only be used when you are registered in the Kerberos database. The *instance* parameter (-i) can be used by a user or for services. The user can enter a null character for the instance or, for administrator privileges, enter *admin.* The service's instance is usually where the host is running the services.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2777E *command* **bad Kerberos instance format**

Explanation: Kerberos cannot recognize the instance you have specified.

System Action: KINIT exits.

User or Operator Response: Re-enter the instance used in the Kerberos database system. For example *admin* should be used if you are a services provider or a remote administrator.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2778I Kerberos realm:

Explanation: This prompt is displayed when the user tries to access the Kerberos database by means of a remote realm.

System Action: KINIT continues.

User or Operator Response: Enter the realm at the appropriate prompt.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2779E *command* **bad Kerberos realm format**

Explanation: Kerberos cannot recognize the realm specified.

System Action: KINIT exits.

User or Operator Response: Re-enter the realm using the correct syntax.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2780I Kerberos ticket lifetime (minutes):

Explanation: This message prompts you for the ticket lifetime in minutes.

System Action: KINIT continues.

User or Operator Response: Enter the lifetime of the ticket in minutes.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2781E *command* **krb_get_lrealm failed**

Explanation: The krb_get_lrealm function, which get the local realm for Kerberos, was unsuccessful.

System Action: KINIT exits.

User or Operator Response: Make sure the local realm is specified in the first line of the *user_id.KRB.CONF* data set.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: krb_get_lrealm

EZB2782I Kerberos realm *realm name*

Explanation: This message states the name of the Kerberos realm. A response will follow indicating the success or failure of your logon attempt.

System Action: KINIT continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: main

EZB2783I error msg.

Explanation: This error message is a indication that Kerberos could not initialize the local realm.

System Action: KINIT continues.

User or Operator Response: Make sure that the local realm is defined in your *user_id.KRB.CONF* data set. If error displays continuously contact the system programmer.

System Programmer Response: Respond to the error message as indicated.

Source Data Set: KINIT

Procedure Name: main

EZB2784E *progname.error msg*

Explanation: The request you have made, using the *KINIT* command, cannot be granted.

System Action: KINIT exits.

User or Operator Response: The local realm must be defined in your *user_id.KRB.CONF* to initialize the specified

realm. If error continues to occur contact the system programmer to clarify the error.

System Programmer Response: Respond to the error message as indicated.

Source Data Set: KINIT

Procedure Name: main

EZB2785E Usage: *command* **{-irvl} {name}**

Explanation: This message illustrates the correct usage of the *KINIT* command. When using the multiple characters, if a space is placed between the characters only the first character will be represented.

System Action: KINIT exits.

User or Operator Response: Reissue the command using the preceding syntax.

System Programmer Response: None.

Source Data Set: KINIT

Procedure Name: usage

Chapter 6. EZB3000—EZB3566

Domain Name Server (DNS) Messages

This section contains Domain Name Server (DNS) messages.

EZB3000I ***** Can't find initialize address for server** *server*: *return_code*

Explanation: The local host was unable to find the address of the Domain Name Server. Without the Domain Name Server, the local host is unable to determine the addresses of other hosts on the network.

System Action: TCPIP continues.

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

System Programmer Response: Add the address of the Domain Name Server to the *hlq*.PROFILE.TCPIP data set. For more information about the *hlq*.PROFILE.TCPIP data set, refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NSLOOKUP

Procedure Name: main

EZB3001I **Usage:**

EZB3002I **nslookup [-opt ...] # interactive mode using default server**

EZB3003I **nslookup [-opt ...] - server #interactive mode using default server**

EZB3004I **nslookup [-opt ...] host #just look up 'host' using default server**

EZB3005I **nslookup [-opt ...] host #just look up 'host' using 'server'**

Explanation: These messages give the format and usage for the NSLOOKUP command. For more information about the NSLOOKUP command, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: Usage

EZB3006I **SetDefaultServer: invalid name:** *name*

Explanation: The name specified for the default Domain Name Server does not correspond to a local host. The Domain Name Server is not initialized.

System Action: TCPIP continues.

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

System Programmer Response: Correct the default Domain Name Server name in the *hlq*.PROFILE.TCPIP data set. For more information about configuring the Domain Name Server, refer to *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: NSLOOKUP

Procedure Name: SetDefaultServer

EZB3007I ***** Can't find address for server** *server*: *reason*

Explanation: TCPIP was unable to change to another Domain Name Server because it could not find an address for the new Domain Name Server. The reason it could not find the address is displayed in the message.

System Action: TCPIP continues.

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

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

Source Data Set: NSLOOKUP

Procedure Name: SetDefaultServer

EZB3008I ***** No** *query_type* **(***query_abbrev.***) records available for** *host*

Explanation: No information of the type requested is available for the indicated host.

System Action: TCPIP continues.

User or Operator Response: Send a PING to the indicated host to determine if it is reachable through the network. Check the *hlq*.SEZAINST data set to make sure that recursion has been requested. Recursion will allow the Domain Name Server for the local zone to communicate with other Domain Name Servers to determine the requested address. If the error persists, notify the system programmer. For more information about recursion, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Make sure that the Domain Name Server is started and online. If necessary, update the address tables for the Domain Name Server to include the indicated host.

Source Data Set: NSLOOKUP

Procedure Name: DoLookup

EZB3009I ***** Request to** *server* **timed-out**

Explanation: A request to the indicated server reached the end of its time to live. The request is not answered.

System Action: TCPIP continues.

User or Operator Response: Resubmit the request. If the error persists, notify the system programmer.

System Programmer Response: Check the indicated server to determine why it is not answering requests.

Source Data Set: NSLOOKUP

Procedure Name: DoLookup

EZB3010I ******* *Server* **can't find** *host*: *reason*

Explanation: The Domain Name Server is unable to find an address for the indicated host. The reason is displayed in the message.

System Action: TCPIP continues.

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

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

Source Data Set: NSLOOKUP

Procedure Name: DoLookup

EZB3011E ***** Can't open** *file* **for writing**

Explanation: The Domain Name Server is unable to open a file to write the answer to a request. The request is not answered.

System Action: TCPIP continues.

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

System Programmer Response: Make sure that the Domain Name Server has the proper write authority.

Source Data Set: NSLOOKUP

Procedure Name: LookupHost, LookupHostWithServer

EZB3012I > *request*

Explanation: This message is written to the output file. It echoes the request to the Domain Name Server that produced the output file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: LookupHost, LookupHostWithServer

EZB3013W * Can't find address for server** *server*: *reason*

Explanation: The default Domain Name Server was unable to get information about the requested Domain Name Server. The reason for the error is displayed in the message. The request is not answered.

System Action: TCPIP continues.

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

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

Source Data Set: NSLOOKUP

Procedure Name: LookupHostWithServer

EZB3014E * Invalid set command**

Explanation: The SET subcommand, which is used to set or change the options for the NSLOOKUP command, was submitted with no associated option. The SET subcommand is not accepted.

System Action: TCPIP continues.

User or Operator Response: If the NSLOOKUP command was submitted in interactive mode, resubmit the command specifying a valid option after the SET subcommand. To accept the current options to the NSLOOKUP command, omit the SET subcommand. If the NSLOOKUP command was submitted from the *user_id*.NSLOOKUP.ENV data set, edit the data set, including a valid option after the SET subcommand to change the NSLOOKUP options, or deleting the SET subcommand to accept the current option settings. For more information on the SET subcommand, the options to the NSLOOKUP command, and the *user_id*.NSLOOKUP.ENV data set, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOption

EZB3015I d2 mode disabled; still in debug mode

Explanation: The *NO D2* option has been accepted, disabling the high-level tracing for the Domain Name Server. The server is still in debug mode.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: SetOption

EZB3016E invalid port value: *port*

Explanation: The PORT option to the SET subcommand, which specifies the port to use when contacting the Domain Name Server, was submitted with an incorrect value for the port. The given value was either out of the range of valid ports, or the value of a port that is already assigned. The PORT option is not accepted.

System Action: TCPIP continues.

User or Operator Response: Resubmit the NSLOOK command, specifying a valid port in the PORT option to the SET subcommand. The Domain Name Server is a well-known service, and is allocated port 53. For more information about the NSLOOKUP command, its subcommands and options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOption

EZB3017E invalid retry value: *value*

Explanation: The RETRY option to the SET subcommand of the NSLOOKUP command was submitted with an incorrect value. The NSLOOKUP command is not accepted. The RETRY option specifies the number of times a request is resent. If the RETRY option is set to 0, no requests are sent, resulting in the error message "no response from server".

System Action: TCPIP continues.

User or Operator Response: Resubmit the NSLOOKUP command with a valid value for the RETRY option of the SET subcommand. For more information about the NSLOOKUP command and its subcommands and options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOption

EZB3018E invalid timeout value: *value*

Explanation: The TIMEOUT option to the SET subcommand of the NSLOOKUP command was submitted with an incorrect value. The NSLOOKUP command is not accepted. The TIMEOUT option specifies the number of seconds to wait before canceling a request. If the TIMEOUT option is set to 0, requests are canceled immediately and are never answered.

System Action: TCPIP continues.

User or Operator Response: Resubmit the NSLOOKUP command with a valid value for the RETRY option of the SET subcommand. For more information about the NSLOOKUP command and its subcommands and options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOption

EZB3019E * Invalid option:** *option*

Explanation: The SET subcommand was submitted with an option that the Domain Name Server did not recognize. The command is not accepted.

System Action: TCPIP continues.

User or Operator Response: Check the spelling and syntax and resubmit the command. For more information about valid options for the SET subcommand, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOption

EZB3020I **Set options:**

EZB3021I *no***debug** *no***defname** *no***search** *no***recurse**

EZB3025I *no***d2** *no***vc** *no***ignoretc port=***port*

EZB3029I **querytype=***type* **class=***class* **timeout=***number* **retry=***number*

EZB3033I **root=***server*

EZB3034I **domain=***domain*

EZB3035I *no***brackets**

EZB3194I *diff/time/no***stamp**

EZB3036I **srchlist=***domain1/domain2/domain3*

Explanation: These messages display the state information used by the resolver library and other options set by the user. The prefix *no* to an option name indicates that the option is not selected. For more information about these options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: ShowOptions

EZB3038E ***** Can't initialize resolver.**

Explanation: The Domain Name Server was unable to initialize the resolver library routines, which are used by clients to request resolution by the Domain Name Server. This indicates that there was an error in specifying the configuration file for the resolver, or the Domain Name Server was unable to allocate the configuration data set. The default name for the configuration data set is *hlq*.RESOLV.CONF.

System Action: Initialization halts. TCPIP continues.

User or Operator Response: Check the syntax of the configuration data set and resubmit the command. If the error persists, specify the configuration options using the NSLOOKUP command and its parameters. For more information about the configuration data set, refer to *OS/390 IBM Communications Server: IP Configuration Reference*. For more information about the NSLOOKUP command and its parameters, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Make more storage available to the Domain Name Server if necessary.

Source Data Set: NSLOOKUP

Procedure Name: Main

EZB3039E ***** Can't find server address for '*server*':**

Explanation: While initializing the resolver library used by the clients to request domain name resolution, the Domain Name Server was unable to find an IP address for the indicated server.

System Action: Initialization halts. TCPIP continues.

User or Operator Response: Correct the *hlq*.RESOLV.CONF data set to include an IP address for the default Domain Name Server. For more information about the *hlq*.RESOLV.CONF data set, refer to *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: ReadRC

EZB3040E ***** Can't find server name for address** *address*: *reason*

Explanation: While initializing the resolver library, which is used by the client to request domain name resolution, the Domain Name Server was unable to find a server name for the indicated address.

System Action: Initialization halts. TCPIP continues.

User or Operator Response: Correct the *hlq*.TCPIP.DATA data set to include a NSINTERADDR statement for the IP address of the Domain Name Server, or enter the NSLOOKUP command, using the *server_name* and *server_address* parameters to specify the default Domain Name Server. For more information about the Domain Name Server configuration and the steps involved, refer to *OS/390 IBM Communications Server: IP Configuration Reference*. For more information about the NSLOOKUP command and its parameters, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: Usage

EZB3041E ***** Default servers are not available**

Explanation: The Domain Name Servers specified as the default servers in the resolver library used by the client to request address resolution by the Domain Name Server are not available.

System Action: Initialization halts. TCPIP continues.

User or Operator Response: Use the NSLOOKUP command with the *server_name* and *server_address* parameters to specify a different default server. If the error persists, notify the system programmer.

System Programmer Response: Make sure that the indicated Domain Name Servers are started and online.

Source Data Set: NSLOOKUP

Procedure Name: Usage

EZB3042I > *command*

Explanation: This message is the command prompt for interactive NSLOOKUP sessions. For more information about interactive sessions using NSLOOKUP, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: Usage

EZB3043I HEADER:

EZB3044I opcode = *code*, **id =** *id*, **rcode =** *return_code*

EZB3047I header flags: *response/query, auth._answer, truncation*

EZB3052I *want recursion, recursion available, primary*

EZB3055I questions = *number*, **answers =** *number*

EZB3057I authority records = *number*, **additional =** *number*

Explanation: These messages display information from the header fields of packets received from a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Fprint_query

EZB3059I QUESTIONS: *server*, **type =** *type*, **class =** *class*

Explanation: This message displays the question records received from name server queries.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Fprint_query

EZB3063I ANSWERS:

EZB3064I —>

Explanation: These messages precede the authoritative answer records generated by a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Fprint_query

EZB3065I AUTHORITY RECORDS:

EZB3064I —>

Explanation: These messages precede the authority records generated by a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Fprint_query

EZB3067I ADDITIONAL RECORDS:

EZB3064I —>

Explanation: These messages precede any additional records generated by a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Fprint_query

EZB3071W (name truncated?)

Explanation: This message indicates a compression error in the resource records generated by a name server query.

System Action: TCPIP continues.

User or Operator Response: Notify the system programmer.

System Programmer Response: Check the indicated Domain Name Server to determine the cause of the compression error. For more information on resource records and the Domain Name Server, refer to *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3072I type = *type*, **class =** *class*, **dlen =** *length*

Explanation: This message is sent to the trace file if the D2 option was specified for the NSLOOKUP command. It displays the type, class, and data length of resource records generated by a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3073I ttl = *number* **(***time***)**

Explanation: This message is sent to the trace file if the DEBUG option has been specified. It indicates the time to live of the table of authority for the Domain Name Server and the time at which the table was last updated in response to a name server query. The authority records for the Domain Name Server are updated at regular intervals for better performance. For more information about authority records for the Domain Name Server, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3078I **address, class =** *number*, **len =** *length*

Explanation: This message is sent to the trace file if the D2 option was specified for the NSLOOKUP command. It displays the class and length of an address type resource record generated by a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3089I **origin =** *name*

EZB3090I **mail addr =** *name*

EZB3091I **serial =** *serial_number* **(***name***)**

EZB3092I **refresh =** *refresh_time* **(***time***)**

EZB3093I **retry =** *retry_time* **(***time***)**

EZB3094I **expire =** *time_to_live* **(***time***)**

EZB3095I **minimum ttl =** *min_time_to_live* **(***time***)**

Explanation: These messages are sent to the trace file if the DEBUG option was specified. They give information about the options selected for the Domain Name Server. For more information on the Domain Name Server, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3097E **errors =** *name*

Explanation: This message indicates the host to which errors will be sent if they are encountered by the Domain Name Server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3105I NULL (dlen *length*)

Explanation: This message is sent to the trace file if the DEBUG option was specified. It indicates that a resource record generated by a name server request has a type of null and the indicated length.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3106E ??? unknown type *type* ???

Explanation: This message is sent to the trace file if the DEBUG option was specified. It indicates that a resource record generated by a name server request has a type that the Domain Name Server does not recognize. The type is displayed in the message.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Check the indicated Domain Name Server to determine why it is generating incorrect resource records.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3107I ttl = *time_to_live* (*time*)

Explanation: This message is sent to the trace file if the DEBUG option was specified. It indicates the time to live for a resource record generated by name server request. Resource records are updated periodically for better performance.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3108E * Error: record size incorrect (*actual_record_size* != *stated_record_size*)**

Explanation: This message is sent to the trace file if the DEBUG option was specified. It indicates that the Domain Name Server received a record that is not equal to the record size set in the *hlq*.PROFILE.TCPIP data set.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: Change the record sizes in the *hlq*.PROFILE.TCPIP data set so that the stated record size matches the actual record size.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3109I SendRequest failed

Explanation: The procedure SendRequest, which sends a request to a name server and returns with the answer, was unsuccessful.

System Action: TCPIP continues.

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

System Programmer Response: Check the Domain Name Server to determine why SendRequest was unsuccessful.

Source Data Set: BD@GET

Procedure Name: GetAnswer

EZB3110I Non-authoritative answer:

Explanation: The name for which the client requested resolution is outside the current Domain Name Server's zone of authority, and recursive resolution was not requested. The Domain Name Server returns the name of the name server most likely to be able to resolve the request. The client issues a request to this name server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@GET

Procedure Name: FreeHostInfoPtr

EZB3111W Size (*size*) too big

Explanation: The size of an answer sent by the Domain Name Server in response to a query by the client is too large to fit into the client's data buffer. The answer cannot be delivered.

System Action: TCPIP continues.

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

System Programmer Response: Increase the size of the client's data buffer using the DATABUFFERPOOLSIZE statement in the *hlq*.PROFILE.TCPIP data set for the client. For more information about the *hlq*.PROFILE.TCPIP data set, refer to *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: BD@GET

Procedure Name: FreeHostInfoPtr

EZB3112I Authoritative answers can be found from: *server(s)*

Explanation: The addresses requested by the client are outside the current Domain name server's zone of authority, and no recursive resolution was requested. The current Domain Name Server returns the names of the servers most likely to have information about the names being requested.

System Action: TCPIP continues.

User or Operator Response: Query the indicated name servers for more information.

System Programmer Response: Assist the user as necessary.

Source Data Set: BD@GET

Procedure Name: FreeHostInfoPointer

EZB3113I Aliased to *host*

Explanation: The address requested by the client is not in the zone of authority for the current Domain Name Server. The name server returns the name, or alias, of the name server most likely to have the address in its domain.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@GET

Procedure Name: GetHostByName

EZB3114I Res_mkquery failed

Explanation: The procedure Res_mkquery, which forms all types of queries and returns the size of the result, was unsuccessful in an attempt to get a server name. The client uses the procedure GetAnswer instead, and if no server name is returned, sets the server name to the default server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@GET

Procedure Name: GetHostInfoByName

EZB3115I res_mkquery() failed

Explanation: The procedure Res_mkquery, which forms all types of queries and returns the size of the result, was unsuccessful in an attempt to get a server name. The client uses the procedure GetAnswer instead, and if no server name is returned, sets the server name to the default server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@GET

Procedure Name: GetHostInfoByAddr

EZB3116E * ls: invalid request** *request*

Explanation: The Domain Name Server received a request it did not recognize. The Domain Name Server is unable to process this request. The request is displayed in this message.

System Action: TCPIP continues.

User or Operator Response: Check the syntax and resubmit the request.

System Programmer Response: Assist the user as necessary.

Source Data Set: BD@LIST

Procedure Name: ListHostsByType

EZB3117I * Can't list domain** *domain***:** *reason*

Explanation: This message is sent to the trace file. The procedure ListHostsByType, which lists the hosts known to the Domain Name Server, was unsuccessful for the indicated reason.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: BD@LIST

Procedure Name: ListHostsByType

EZB3118E * ls: res_mkquery failed**

Explanation: The function res_mkquery, which was called to create a query packet for the requested domain name, was unable to create the query packet. Without a query packet, the LS command, used to list the hosts known to the Domain Name Server for a given domain, cannot be processed.

System Action: TCPIP continues.

User or Operator Response: Check the syntax and resubmit the command. If the error persists, notify the system programmer.

System Programmer Response: Check the trace file for indications of the reason res_mkquery was unable to create a query packet for the indicated command. Possibilities include insufficient buffer space and an incorrect or missing domain name.

Source Data Set: BD@LIST

Procedure Name: ListHostsByType

EZB3119I **[*host*]**

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

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ListHostsByType

EZB3120E ***** Can't open *file* for writing**

Explanation: The function OpenFile, which parses the command string for a file name and opens the file for writing, was unsuccessful, indicating an incorrect command string or an error opening the file. The data set is not opened.

System Action: TCPIP continues.

User or Operator Response: Check the syntax and resubmit the command. If the error persists, notify the system programmer.

System Programmer Response: Make sure that the user has the proper authority to write the indicated data set.

Source Data Set: BD@LIST

Procedure Name: ListHosts

EZB3121I >*command*

Explanation: This message is written to a file opened by the function OpenFile. It echoes the last command submitted by the user.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ListHosts

EZB3122I **Alias**

Explanation: This message precedes the alias of a host. Hosts can be addressed by their alias, their network address, or their canonical name. For more information about aliases, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ListHosts

EZB3124I Received *number* **records.**

Explanation: The Domain Name Server received the indicated number of records in response to a query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ListHosts

EZB3125E * ls: error receiving zone transfer:**

EZB3126I result: *result*, **answers =** *number*, **authority =** *number*, **additional =** *number*

Explanation: The client was unable to read the response from an LS command, used to list the name servers in other domains known to the local Domain Name Server. The data from the LS command is not transferred.

System Action: TCPIP continues.

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

System Programmer Response: Check the Domain Name Server to make sure that it is properly configured to be compatible with the other Domain Name Servers on the system. For more information on configuring the Domain Name Server, refer to *OS/390 IBM Communications Server: IP Configuration Reference*.

Source Data Set: BD@LIST

Procedure Name: ListHosts

EZB3127I *domain_name*

Explanation: This message displays the domain name for which an answer returned from the name server is valid.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3128I result: *result*, **answers =** *number*, **authority =** *authority*, **additional =** *additional_responses*

Explanation: This message displays the results of a search by the Domain Name Server. The message displays the result code from the search, the number of answers received and the zone of authority from which they were received, and the number of additional, nonauthoritative answers received.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: Main.

EZB3131I **(***address_protocol_identifiers*

Explanation: This message indicates the internet address protocol being used by NSLOOKUP. For more information about internet address protocols, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3132I **(dlen =** *length***?)**

Explanation: This message indicates the length of data in a response received by NSLOOKUP.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3133I ***

Explanation: This message precedes other informational messages.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3134I *server, type class*

Explanation: This message displays the name of the server being queried, the type of query, and the class of address being requested.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_cdname_sub

EZB3143I *text data*

Explanation: This is variable data which is displayed in response to the NSLOOKUP command. It is defined by the installation in the name server data base.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3144I *number*

Explanation: This message displays the user or group ID from a name server query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3145I *number*

Explanation: This message displays the protocol value taken from a WKS resource record. For more information about resource records, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3146 *protocol*

Explanation: This message displays the mnemonic for the protocol value taken from a well known services (WKS) resource record. For more information about resource records, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: PrintListInfo

EZB3150E ***** Can't open** *data_set* **for reading**

Explanation: The Domain Name Server was unable to read from the indicated data set. The Domain Name Server query is not answered.

System Action: TCPIP continues.

User or Operator Response: Make sure the indicated data set is in storage accessible to the Domain Name Server.

System Programmer Response: Assist the user as necessary.

Source Data Set: BD@LIST

Procedure Name: ViewList

EZB3151I *host*

Explanation: This message displays the output of the LS function.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ViewList

EZB3154I Finger: no current host defined.

Explanation: No host has been defined to the current finger server, which displays information about users of a remote host. The FINGER query is not answered.

System Action: TCPIP continues.

User or Operator Response: Resubmit the FINGER command, using the format FINGER@host to define the remote host to be used to complete the request.

System Programmer Response: Assist the user as necessary.

Source Data Set: BD@LIST

Procedure Name: Finger

EZB3155I Finger: unknown service

Explanation: The finger server, which gives information about the users of a remote host, received a request for a service that it does not recognize. The request is not processed.

System Action: TCPIP continues.

User or Operator Response: Check the syntax and resubmit the command. For more information about the finger server, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: BD@LIST

Procedure Name: Finger

EZB3156I > *data_set_name*

Explanation: This message displays the name of the data set to which information returned by the finger server will be sent.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: Finger

EZB3157I SendRequest(), len *message_length*

Explanation: The procedure SendRequest, which sends a request packet to the name server and returns with the answer, received an answer of the indicated length.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3158I res_send()

Explanation: This message is displayed if the DEBUG option has been specified. It indicates that the client is using the procedure res_send to send a resolver query packet.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3159I Querying server (*number*) address = *address*

Explanation: The client is sending a request to the indicated name server. The number displayed in parentheses indicates the number of times the server has been queried.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3160I response truncated

Explanation: The reply sent by a name server is larger than the buffer allocated for it. The client dynamically allocates more storage to receive the reply.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendReply

EZB3161I timeout (*number* secs)

Explanation: A query sent to a name server has not been answered within the indicated time. The connection is closed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3162I Old answer: *answer*

Explanation: The indicated answer is an old answer from the Domain Name Server, and it should be ignored.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3163I old answer: *answer*

Explanation: This message is sent to the trace file if the DEBUG option has been specified. It indicates that the given answer from the Domain Name Server is not current and should be ignored.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3164I truncated answer

Explanation: An answer sent from the Domain Name Server was too large to fit in the buffer allocated for it. The resolver will get the rest of the answer using the TCP layer for transport, which allows large answers to be fragmented into manageable packets.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3165I Got answer (*number* **bytes):** *answer*

Explanation: The resolver received an answer of the indicated length to a query sent to the Domain Name Server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3166I Got answer: *answer*

Explanation: The resolver received the indicated answer from a query to the Domain Name Server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3167I got answer: *answer*

Explanation: This message is sent to the trace file if the DEBUG option was specified. It indicates that the resolver received the indicated answer from a query to the Domain Name Server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SEND

Procedure Name: SendRequest

EZB3169E ***** Can't allocate memory**

Explanation: The Domain Name Server was unable to allocate storage to hold lookup tables because no data set name was specified.

System Action: TCPIP continues.

User or Operator Response: Check the syntax and resubmit the command. If the error persists, notify the system programmer.

System Programmer Response: Make sure that sufficient storage is available for the tables needed by the Domain Name Server. If the error persists, check for allocation errors on the server.

Source Data Set: BD@SUBR

Procedure Name: Malloc

EZB3170I **Server:** *server*

EZB3171I **Addresses:** *addresses*

EZB3172I **Address:** *address*

Explanation: These messages give the name and addresses of the server being queried.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SUBR

Procedure Name: PrintHostInfo

EZB3175I **Aliases:** *alias_names*

Explanation: This message is displayed with message EZB3170I and gives the alias names corresponding to the indicated addresses.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SUBR

Procedure Name: PrintHostInfo

EZB3176I **Served by:**

EZB3177I *servers*

EZB3178I *, internet_address*

Explanation: These messages are displayed with message EZB3170I. They list the name servers for the indicated addresses.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@SUBR

Procedure Name: PrintHostInfo

EZB3181W unknown query class: *class*

Explanation: The Domain Name Server received a query of a class it does not recognize. Repeated queries of this class will tie up the name server, preventing it from replying to valid queries. The query is not processed.

System Action: TCPIP continues.

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

System Programmer Response: Check the client to determine why it is sending incorrect queries. Take the client offline until the problem is solved.

Source Data Set: BD@SUBR

Procedure Name: StringToClass

EZB3182W unknown query type: *type*

Explanation: The Domain Name Server received a query of a type it does not recognize. The query cannot be processed, and too many queries of this type will tie up the name server, preventing it from replying to valid queries. The query is not processed.

System Action: TCPIP continues.

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

System Programmer Response: Check the indicated client to determine why it is sending queries of an incorrect type. Take the client offline until the problem is corrected.

Source Data Set: BD@SUBR

Procedure Name: StringToType

EZB3194I *type***stamp**

Explanation: This message indicates the type of time stamping selected for the NSLOOKUP command. The possible values are:

Value	Indicates
time	The time is displayed before each output line.
diff	The time is displayed before each output line only when the time changes.
no	No time stamping is selected.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: ShowOptions

EZB3195I [*host***]**

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

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ListHostsByType

EZB3196W no DOMAIN is entered, using default

Explanation: The Domain Name Server received a command that does not specify a domain. The name server uses the default domain.

System Action: TCPIP continues.

User or Operator Response: Reenter the command, specifying the correct domain if necessary.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOptions

EZB3197W no ROOT is entered, using default

Explanation: The Domain Name Server received a command that did not specify the root name server to be used. The Domain Name Server uses the default server.

System Action: TCPIP continues.

User or Operator Response: Resubmit the command, specifying the root server if necessary.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOptions

EZB3198W no SRCHLIST is entered, using default

Explanation: The Domain Name Server received a command that did not specify the search list to be used. The Domain Name Server uses the default search list.

System Action: TCPIP continues.

User or Operator Response: Resubmit the command, specifying the search list if necessary.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: SetOptions

EZB3201I text = *text*

Explanation: This message displays the contents of a text type query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@DEBUG

Procedure Name: Print_rr

EZB3202E * Syntax error in** *option* **option.**

Explanation: The indicated option of the NSLOOKUP command was submitted with incorrect syntax. The option is ignored.

System Action: TCPIP continues.

User or Operator Response: Resubmit the NSLOOKUP command with the proper syntax for all options. For more information about the NSLOOKUP command and the proper syntax for its options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: comml_option

EZB3203E ***** The '=' sign is missing in** *option* **option**

Explanation: The indicated option of the NSLOOKUP command was submitted without a '='. The option cannot be processed.

System Action: TCPIP continues.

User or Operator Response: Resubmit the NSLOOKUP command with the proper syntax for all options. For more information about the NSLOOKUP command and its options, see the *OS/390 IBM Communications Server: IP User's Guide*.

System Programmer Response: Assist the user as necessary.

Source Data Set: NSLOOKUP

Procedure Name: comml_option

EZB3205I *local host*

Explanation: This is the name of the local host or the name of a server that knows about the local domain.

System Action: NSLOOKUP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: BD@LIST

Procedure Name: ListSubr

EZB3206I **nslookup [-option ...] [host-to-find | - [server]]**

EZB3207I **Commands: (identifiers are shown in uppercase, <> means optional)**

EZB3208I **NAME - print info about the host/domain NAME using**

EZB3209I **default server**

EZB3210I **NAME1 NAME2 - as above, but use NAME2 as server**

EZB3211I **help or ? - print info on common commands;**

EZB3212I **use TSO's help nslookup for more details**

EZB3213I **set OPTION - set an option**

EZB3214I **all - print options, current server and host**

EZB3215I **<no>debug - print debugging information**

EZB3216I	**<no>d2 - print exhaustive debugging information**
EZB3217I	**<no>defname - append domain name to each query**
EZB3218I	**<no>recurse - ask for recursive answer to each query**
EZB3219I	**<no>vc - always use a virtual circuit**
EZB3220I	**domain=NAME - set default domain name to NAME**
EZB3221I	**srchlist=N1</N2/.../N6> - set domain to N1 and search**
EZB3222I	**list to N1,N2, etc.**
EZB3223I	**root=NAME - set root server to NAME**
EZB3224I	**retry=X - set number of retries to X**
EZB3225I	**timeout=X - set initial time-out interval to X seconds**
EZB3226I	**querytype=X - set query type, e.g.,**
EZB3227I	**A,ANY,CNAME,HINFO,MC,NS,PTR,SOA,WKS**
EZB3228I	**type=X - synonym for querytype**
EZB3229I	**class=X - set query class to one of IN (Internet),**
EZB3230I	**CHAOS, HESIOD or ANY**
EZB3231I	**server NAME - set default server to NAME, using current**
EZB3232I	**default server**
EZB3233I	**lserver NAME - set default server to NAME, using initial server**
EZB3234I	**finger <USER> - finger the optional NAME at the current**
EZB3235I	**default host**
EZB3236I	**root - set current default server to the root**
EZB3237I	**ls <opt> DOMAIN [> DATASET] - list addresses in DOMAIN**

EZB3238I (optional: output to DATASET)

EZB3239I -a - list canonical names and aliases

EZB3240I -h - list HINFO (CPU type and operating system)

EZB3241I -s - list well-known services

EZB3242I -d - list all records

EZB3243I -t TYPE - list records of the given type

EZB3244I (e.g., A, CNAME, MX, etc.)

EZB3245I view DATASET - sort an 'ls' output file and view it with more

EZB3246I exit - exit the program

Explanation: This is the summary of available commands and options that you can review by typing help or ? at the > prompt (see message EZB3042I), once you have entered the NSLOOKUP interactive session. For details about any of these commands or options, see *OS/390 IBM Communications Server: IP User's Guide*.

System Action: The name server continues until you type `exit` at the > prompt (see message EZB3042I).

User or Operator Response: Use these help messages as a guide for entering subsequent NSLOOKUP commands.

System Programmer Response: None.

Source Data Set: NSLOOKUP

Procedure Name: PrintHelp()

EZB3250I invalid dig option *option*

Explanation: The option indicated was entered at the command line and is not a valid DIG option.

System Action: TCPIP continues.

User or Operator Response: Enter a valid DIG option. *OS/390 IBM Communications Server: IP User's Guide* contains the DIG command syntax and the valid options.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3251I no dig -T value specified

Explanation: No wait time (-T value) was specified. Wait time is the time to wait between successive queries when operating in batch mode. The default wait time is 0, which indicates no wait time.

System Action: TCPIP continues.

User or Operator Response: If you wish to specify a wait time other than 0, issue the DIG command with the -T option. See *OS/390 IBM Communications Server: IP User's Guide* for the syntax of the DIG command. Otherwise, no action is necessary.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3252I invalid dig -T value *tvalue*

Explanation: The -T value specified on the DIG command is not valid.

System Action: TCPIP continues.

User or Operator Response: Specify a -T value in seconds. 0 is the default.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3253I ; invalid class specified

Explanation: The network class specified on the DIG command is not valid.

System Action: TCPIP continues.

User or Operator Response: Specify a valid network class on the DIG command. DIG recognizes only the IN, CHAOS, HESIOD, and ANY network classes.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3254I ; invalid type specified

Explanation: The query type specified on the DIG command is not valid.

System Action: TCPIP continues.

User or Operator Response: Specify a valid query type, *qtype* on the DIG command. See *OS/390 IBM Communications Server: IP User's Guide* for information about the DIG command and valid query types.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3255I Missing batch file name

Explanation: The -f option was specified on the DIG command with no data set name.

System Action: TCPIP continues.

User or Operator Response: Specify a batch data set name on the -f option of the DIG command and reissue the command. *OS/390 IBM Communications Server: IP User's Guide* contains information about the DIG command and -f option.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3256I invalid dig -x option *option*

Explanation: The -x option specified on the DIG command is not valid.

System Action: TCPIP continues.

User or Operator Response: Specify a valid dotted decimal notation IP address. *OS/390 IBM Communications Server: IP User's Guide* describes the DIG command and the -x option.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3257I no dig -p value specified

Explanation: The DIG command -p option was specified with no value. The port number given when contacting the name server must be specified with the -p option.

System Action: TCPIP continues.

User or Operator Response: Specify the port number given when contacting the name server on the DIG command with the -p option. The default for the Domain Name Server is 53. The -p option allows you to override this default.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3258I invalid dig -p value *pvalue*

Explanation: The -p value specified on the DIG command is not valid.

System Action: TCPIP continues.

User or Operator Response: Specify a valid port number on the DIG -p option and reissue the command. Use the decimal port number given when contacting the name server.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3259I ; invalid type/class specified

Explanation: The query type or query class specified on the DIG command is not valid.

System Action: TCPIP continues.

User or Operator Response: Specify a valid query class and query type on the DIG command. See *OS/390 IBM Communications Server: IP User's Guide* for information about valid classes and types.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3260I ; pflag: *pfcode* **res:** *resvalue*

Explanation: This message displays the print flag values and the resolver options that are set for the request.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3262I ; Bad server: *servername* **- - using default server and timer opts**

Explanation: The server specified is not a recognized server. DIG is using the default server and the associated timer options instead.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3263I ;; FROM: *hostname* **to SERVER:** *servername*

Explanation: Messages are being sent from the host indicated to the server indicated.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3264I ;; WHEN: *time*

Explanation: The day, date, and time of the DIG request is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3265I ;; re_mkquery: buffer too small

Explanation: The storage buffer is too small to complete the query request.

System Action: TCPIP continues.

User or Operator Response: Define a larger buffer and reissue the DIG query request.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3266I ;; Error time:

Explanation: This message indicates the time that elapsed before the resolver encountered an error.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3267I Sent *number* **pkts, answer found in time:** *msecond*

Explanation: The indicated number of packets was transferred between the host and the server. Also included in the message is the time it took to find the number of packets sent.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3268I sent *number* **too many pkts**

Explanation: The resolver sent too many query packets to a name server. The number of packets is indicated in the message.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3271I ;; MSG SIZE sent *sendsize* **rcvd:** *recvsize*

Explanation: The size of the message when sent from the host and as received by the server is displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3272I ; ** unexpected answer type** *code***,size** *size*

Explanation: The answer type encountered by DIG is not recognized. Valid answer (query) types can be found in *OS/390 IBM Communications Server: IP User's Guide*.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@GETH

Procedure Name: getanswer

EZB3273I ; ** size (*size***) too big**

Explanation: The information requested is too large for the answer buffer. This message is issued when debugging is requested.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@GETH

Procedure Name: getaddress

EZB3274I ; ** res_search failed

Explanation: The RES@SEAR routine did not complete successfully. This message is issued when debugging is requested and this error occurs.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: DIG@GETH

Procedure Name: gethostbyname

EZB3275I *number.number.number.number* **in-addr/arpa**

Explanation: This message indicates the dotted decimal domain name that is being queried.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@GETH

Procedure Name: gethostbyaddr

EZB3276I ;; ** res_query failed

Explanation: The RES@QUER routine did not complete successfully. This message is issued when debugging is issued and this error occurs.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: DIG@GETH

Procedure Name: gethostbyaddr

EZB3279I ; ***zone transfer: res_mkquery failed

Explanation: The RES@MKQU routine did not complete successfully during a zone transfer. This message is issued when debugging is requested and this error occurs.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: DIG@LIST

Procedure Name: do_zone

EZB3280I ; Matching SOA found

Explanation: The authority record being searched for was located.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@LIST

Procedure Name: do_zone

EZB3281I ; ListHosts: error receiving zone transfer:

Explanation: An error occurred during a zone transfer between a primary and a secondary name server.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: DIG@LIST

Procedure Name: do_zone

EZB3282I ; result: *text*, **answers =** *value*, **authority =** *value*

Explanation: This message displays with message EZB3281I and indicates the result of the zone transfer attempt, information from the answer section of the response, and the address of the authoritative name server for the response.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@LIST

Procedure Name: do_zone

EZB3283X additional= *number*

Explanation: This message appears with message EZB3282I. Additional resources records that have not been requested, but might be useful, are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@LIST

Procedure Name: do_zone

EZB3284I ; ***Error during listing of *domain*

Explanation: An error occurred during the listing of information about the domain name indicated in the message.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: DIG@LIST

Procedure Name: do_zone

EZB3285X *number* **msec**

Explanation: This message displays the number of microseconds that the request took to complete.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@QTIM

Procedure Name: prnttime

EZB3286I ;* Invalid option:** *option*

Explanation: A DIG command option that was specified is not valid.

System Action: TCPIP continues.

User or Operator Response: Check the command you entered to be sure you specified the command with the correct syntax. *OS/390 IBM Communications Server: IP User's Guide* describes the DIG command and its syntax.

System Programmer Response: None.

Source Data Set: DIG@OPT

Procedure Name: SetOption

EZB3287I invalid timeout value: *value*

Explanation: The time-out value specified on the DIG command is not a valid value. The time-out value specifies the number of seconds to wait before timing out of a request.

System Action: TCPIP continues.

User or Operator Response: Specify a decimal digit value for the time-out value using the DIG command.

System Programmer Response: None.

Source Data Set: DIG@OPT

Procedure Name: SetOption

EZB3288I invalid retry value: *limit*

Explanation: The retry limit specified on the DIG command is not valid. The retry value specifies the number of times a request is sent.

System Action: TCPIP continues.

User or Operator Response: Specify a decimal digit value for the retry limit using the DIG command.

System Programmer Response: None.

Source Data Set: DIG@OPT

Procedure Name: SetOption

EZB3289I ; ***Bad char in numeric string -- ignored

Explanation: DIG found a nonnumeric character in a numeric string. DIG is ignoring the character.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@OPT

Procedure Name: SetOption

EZB3290I ; ** unknown query class: *class*

Explanation: The specified query class is not valid. *OS/390 IBM Communications Server: IP User's Guide* lists and describes the valid query classes.

System Action: TCPIP continues.

User or Operator Response: Specify a valid query class and reissue the DIG command.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: StringToClass

EZB3291I ; ** unknown query type: *type*

Explanation: The specified query type is not valid. *OS/390 IBM Communications Server: IP User's Guide* lists and describes the valid query types.

System Action: TCPIP continues.

User or Operator Response: Specify a valid query type and reissue the DIG command.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: StringToType

EZB3293I ; ** Calloc failed

Explanation: The Calloc subroutine did not complete successfully.

System Action: TCPIP continues.

User or Operator Response: None.

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

Source Data Set: DIG@SUBR

Procedure Name: Calloc

EZB3294I *title word*

Explanation: This is the title and the host information structure name that is printed when DIG prints the host information structure.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3295I Addresses:

Explanation: These are the addresses contained in the host information structure for the host.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3296I Address:

Explanation: This is the address contained in the host information structure for the host.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3298I *charword stringterm*

Explanation: These fields contain additional data from the host information structure.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3299I Aliases:

Explanation: The host aliases are displayed following this message.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3300I Served by:

Explanation: The server names are displayed following this message.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3301I - *servername*

Explanation: This message indicates the server name.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SUBR

Procedure Name: PrintHostInfo

EZB3304I ;; Querying server ("*server***") address =** *addr*

Explanation: This indicates the server name and the address of the server to which a request is being sent.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SEND

Procedure Name: res_send

EZB3305I ;; id = *idname* **- sending now:**

Explanation: This messages contains the name of the host id.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SEND

Procedure Name: res_send

EZB3307I socket is *state*

Explanation: This message indicates whether the socket to which a connection is being established is ready or not ready.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SEND

Procedure Name: res_send

EZB3309I ;; timeout at:

Explanation: This message gives the time at which a time-out occurred while a server request was being sent.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response:

Source Data Set: DIG@SEND

Procedure Name: res_send

EZB3310I ;**; Old Answer id:** *idname*

Explanation: This message indicates the host name to which a query response is to be returned. The request is not current. This message is issued when debugging is requested.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SEND

Procedure Name: res_send

EZB3311I ;; truncated answer

Explanation: The answer returned from the queried name server, is not complete. This message is issued when debugging is turned on.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@SEND

Procedure Name: res_send

EZB3312I ;; res_mkquery(*opcode domainname qclass qtype***)**

Explanation: This message displays the operation type, DNS server name, the query class, and the query type of the server query. This message is issued when debugging is requested.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@MKQU

Procedure Name: res_mkquery

EZB3314X *service_name*

Explanation: This message displays the service name. This message is issued for debugging purposes.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3316I cms;; ->>HEADER<<-

EZB3317X opcode: *code*

EZB3318X , status *status*

EZB3319X , id: *idname*

Explanation: These messages display in response to a DIG command. They indicate the operation type, for example, QUERY; the status of the request, for example, noerror, which says that the request was made correctly; and the ID of the queried Domain Name Server. If the DIG command is issued with no other parameters, this information is returned about the default server.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: fp_query

EZB3320I ;; flags:

EZB3321X qr

EZB3322X aa

EZB3323X **tc**

EZB3324X **rd**

EZB3325X **ra**

EZB3326X **pr**

EZB3327X **res_opts:** *options*

Explanation: Some combination of the above flags are displayed to indicate which options are set on. The flags and their meanings are as follows:

Flag	Meaning
qr	Prints the outgoing query.
aa	Accepts only authoritative responses to queries.
tc	Truncated.
rd	Recursion desired.
ra	Recursion available.
pr	Uses only the primary name server for the zone.
res_opts	resolver options

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response:

Source Data Set: DIG@DEBU

Procedure Name: fp_query

EZB3328I **;; Ques:** *value***,**

EZB3329I **Ans:** *value***,**

EZB3330I **Auth:** *value***,**

EZB3331I **Addit:** *value***,**

Explanation: This message displays the values of the question, answer, authoritative, and additional sections of the response. The question section contains the original query; the answer section contains the set of all resource records from the name server database that satisfy the query; the authoritative section contains resource records that specify the address of an authoritative name server for the query; and the additional section contains resource records that have not been explicitly requested, but could be useful.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: fp_query

EZB3332I **;; QUESTIONS:**

EZB3333X **;;**

EZB3334X **, type =** *type*

EZB3335X **, class =** *class*

Explanation: This messages prints the question records as well as the type of query requested and the network class requested.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3336X *timetolive*

Explanation: This is the time-to-live (TTL) value for the resource record. TTL is the number of seconds that a record is valid in a cache.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3337I *class type*

Explanation: This is the network class requested in the query and the type of query to be performed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3338X *type*

Explanation: This is the type of query to be performed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3339I **;; proto:** *number*

EZB3340X **, port:** *number*

Explanation: These messages indicate the network protocol and the port number for the domain name identified by the address record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3341X *string*

EZB3342X *string*

Explanation: These messages indicate the central processing unit type and the operating system of a node. This information is part of the host information record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response:

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3343X **(***serialvalue* **;serial**

EZB3344X *refreshvalue* **;refresh**

EZB3345X *retryvalue* **;retry**

EZB3346X *expirevalue* **;expire**

EZB3347X *minimvalue***) ;minim**

Explanation: These messages are part of the authoritative section of the response to a DIG query that requested authority records. The following list defines each of the values:

- Serial is the serial number of the zone database.
- Refresh is the refresh interval, or the length of time, in seconds, you must allow between the refreshing of a database from a remote name server.
- Retry is the retry interval that indicates the length of time, in seconds, you must allow before retrying a failed refresh.
- Expire is the expiration time-to-live that indicates the maximum time for records to be valid in the zone database.
- Minim is the minimum time-to-live that indicates the minimum time for records to be valid in the zone database.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3348X *term*

Explanation: This message identifies a host that can act as a mail exchange for the domain specified in the domain name field. A mail exchange runs a mail agent that delivers or forwards mail for the domain name specified in the first field of the resource record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3349X *term*

Explanation: This message displays the group ID associated with the resource record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3350X *string string* **(**

Explanation: This message indicates the IP address and protocol names for the resource record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3351X *string number* **(**

Explanation: This message indicates the protocol name and number for the resource record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3352I *number*

Explanation: This message indicates the number of protocol numbers for services stored in the well-known services record.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3353X First *number* bytes of hex data:

Explanation: This message displays the number of bytes that are displayed following this message. This message is issued when debugging is requested.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3354I *data*

Explanation: This is the data in a resource record that appears in an unspecified format (binary format). Message EZB3353X tells the number of bytes of the data that are displayed.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3355X ; *deftimetolive*

Explanation: This is the default time-to-live (TTL) value. TTL is the number of seconds that a record is valid in a cache.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3356I ;; packet size error

Explanation: The query packet is not the standard size.

System Action: TCPIP continues.

User or Operator Response: Resubmit the query request.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3359I ;; ANSWERS:

Explanation: This message precedes the display of records in the answers section of a query response.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: fp_query

EZB3360I ;; AUTHORITY RECORDS:

Explanation: This message precedes the display of records in the authoritative section. These resource records specify the address of an authoritative name server for the query.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: fp_query

EZB3361I ;; ADDITIONAL RECORDS:

Explanation: This message precedes the display of records in the additional section of a query response.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: fp_query

EZB3362I ; <<>>DiG *version* **<<>>**

Explanation: This messages displays the version of the Domain Information Groper (DIG) currently in use on the system.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: main

EZB3363X *defwell-known-service*

Explanation: The service displayed is the default well-known-service name.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG@DEBU

Procedure Name: p_rr

EZB3364I dig.help

Explanation: The help command has been issued and the DIG.HELP data set is being opened.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp

EZB3365I file open

Explanation: The help data set is now open.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: main

EZB3366I - August 30, 1990

EZB3367I - dig *@server* **domain** *query-type query-class*

EZB3368I *+query-option –dig-option comment*

EZB3369I server - a domain name or a dot-notation Internet address

EZB3370I default: nsinteraddr parm address defined in

EZB3371I dataset TCPIP.DATA

EZB3372I DiG Defaults: Name server address in DIG.ENV dataset

EZB3373I domain - request domain information for domain name

EZB3374I query-type - a, any, cname, hinfo, mx, ns, ptr, soa, wks

EZB3375I query-class - IN (Internet), CHAOS(obsolete), HESIOD

EZB3376I *+query option*

EZB3377I <no>debug (deb) turn on/off debugging mode <deb>

EZB3378I <no>d2 turn on/off extra debugging mode <nod2>

EZB3379I <no>recurse (rec) use/don't use recursive lookup <rec>

EZB3380I retry=# (ret) set number of retries to # <4>

EZB3381I time=# (ti) set timeout length to # seconds <4>

EZB3382I <no>ko keep open option (implies vc) <noko>

EZB3383I <no>vc use/don't use virtual circuit <novc>

EZB3384I	**<no>defname use/don't use default domain name <novc>**
EZB3385I	**<no>search (sea) use/don't use domain search list <sea>**
EZB3386I	**domain=NAME (do) set default domain name to NAME**
EZB3387I	**<no>ignore (i) ignore/don't ignore trunc. errors <noi>**
EZB3388I	**<no>primary (pr) use/don't use primary server <nopr>**
EZB3389I	**<no>aaonly (aa) authoritative query only flag <noaa>**
EZB3390I	**<no>sort (sor) sort resource records <nosor>**
EZB3391I	**<no>cmd echo parsed arguments <cmd>**
EZB3392I	**<no>stats (st) print query statistics (RTT,etc) <st>**
EZB3393I	**<no>Header (H) print basic header <H>**
EZB3394I	**<no>header (he) print header flags <he>**
EZB3395I	**<no>ttlid (tt) print TTLs <tt>**
EZB3396I	**<no>cl print class info <nocl>**
EZB3397I	**<no>qr print outgoing query <noqr>**
EZB3398I	**<no>reply (rep) print reply <rep>**
EZB3399I	**<no>ques (qu) print question section <qu>**
EZB3400I	**<no>answer (an) print anwer section <an>**
EZB3401I	**<no>author (au) print authoritative section <au>**
EZB3402I	**<no>addit (ad) print additional section <ad>**
EZB3403I	**pfdef set to default print flags**
EZB3404I	**pfmin set to minimal default print flags**
EZB3405I	**pfset=# set print flags to #**

EZB3406I	**(# can be hex/octal/decimal)**
EZB3407I	**pfand=# bitwise and print flags with #**
EZB3408I	**pfor=# bitwise or print flags with #**
EZB3409I	**-** *dig option*
EZB3410I	**–x dot-notation-address**
EZB3411I	**–f file for dig batch mode**
EZB3412I	**–T Time in seconds between start of successive queries**
EZB3413I	**–p Port number**
EZB3414I	**–P** *ping-string*
EZB3415I	**–t query-type**
EZB3416I	**–c query-class**
EZB3417I	**–envsav This flag specifies that the dig environment**
EZB3418I	**(defaults, print options, etc.), after all of**
EZB3419I	**the arguments are parsed, should be saved to a**
EZB3420I	**file to become the default environment. DiG.env**
EZB3421I	**is created in the current working directory.**
EZB3422I	**–envset This flag only affects batch query runs. When**
EZB3423I	**–envset is specified on a line in a dig batch file**
EZB3424I	**the dig environment after the arguments are parsed,**
EZB3425I	**becomes the default environment for the duration of**
EZB3426I	**the batch file, or until the next line which**
EZB3427I	**specifies –envset.**

EZB3428I **–<no>stick This flag only affects batch query runs. It**

EZB3429I **specifies that the dig environment (as read**

EZB3430I **initially or set by –envset switch) is to be**

EZB3431I **restored before each query (line) in a dig**

EZB3432I **batch file.**

EZB3433I **%comment - included argument that is not parsed**

Explanation: Messages EZB3364I - EZB3430I appear in response to the HELP DIG command. See *OS/390 IBM Communications Server: IP User's Guide* for more information about the DIG command.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: DIG

Procedure Name: PrintHelp()

EZB3483E **Couldn't open output file**

Explanation: NSDBLOAD encountered an error opening its output file. The program ends with exit code 99.

System Action: The name server exits.

User or Operator Response: Check that the name server has read/write access to the file.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: main

EZB3484E **Error(s) occurred reading master datafile.**

Explanation: NSDBLOADz encountered an error while processing the master data file.

System Action: The name server ends.

User or Operator Response: Check the data syntax of the master data file.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NSDBLOAD

Procedure Name: main

EZB3487I **Error occurred deleting table info.**

Explanation: NSDBLOADz encountered an error deleting SQL table information.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: main

EZB3489I *sql statement* **sqlcode** *sqlcode*

Explanation: NSDBLOAD was unable to process the indicated SQL command.

System Action: NSDBLOAD continues.

User or Operator Response: Use the *sqlcode* that is displayed in this message and *DB2 Messages and Codes* to determine the cause of the error and respond as indicated.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name:

EZB3490I *time*: **Delete all data from** *name* **table**

Explanation: NSDBLOADz is deleting all data from the indicated table.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: deleteinfo

EZB3491I *time*: **Deleted all data in sql table** *name*

Explanation: NSDBLOADz has deleted all data in the indicated SQL table.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: deleteinfo

EZB3493E Couldn't read output resource record file

Explanation: NSDBLOADz encountered an error opening its output file. The program ends with exit code 99.

System Action: The NSDBLOAD program ends.

User or Operator Response: Check for the name server's access to the file.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3495W error invalid class *class*

Explanation: The NSDBLOAD program encountered an incorrect class specification in an input record. The record is ignored.

System Action: TCPIP continues.

User or Operator Response: Check for the correct CLASS of the defined resource record.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3496W error invalid type *type*

Explanation: The NSDBLOAD program encountered an incorrect type specification in an input record.

System Action: The record is ignored.

User or Operator Response: Check for the correct TYPE of the defined resource record.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3497I *time*: **Finished adding data to table ....**

Explanation: The NSDBLOAD program has finished adding data to its SQL table.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3498I *time*: **Update statistics on the** *name* **table**

Explanation: The NSDBLOAD program updates the statistics on this table.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3499I *time*: **Data base update completed.**

Explanation: The NSDBLOAD program has completed the database update.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3501E Couldn't read input master file

Explanation: The NSDBLOAD program encountered an error opening the resource record file previously generated. The program ends with exit code 99.

System Action: The name server ends.

User or Operator Response: Check the read/write access on the input file for the current user ID.

System Programmer Response: Check for the existence of a resource record file.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3503E Origin definition error

Explanation: The NSDBLOAD program encountered a $ORIGIN statement in the input file without an accompanying definition.

System Action: TCPIP continues.

User or Operator Response: Correct the $ORIGIN input statement to NSDBLOAD.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3504I *time*: **New origin** *origin*

Explanation: The NSDBLOAD program encountered a $ORIGIN statement with the specified origin in its input file.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3505I *time*: **@ - current origin** *origin*

Explanation: The NSDBLOAD program is using the specified origin.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3506E Origin has not been defined

Explanation: A relative domain name defined in a master file requires a domain origin suffix.

System Action: TCPIP continues.

User or Operator Response: Correct the master file to include a $ORIGIN *domain* statement.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3509W Format error 1: *line*

Explanation: The NSDBLOAD program encountered a syntax error in the input file. The line causing the error is displayed and ignored.

System Action: TCPIP continues.

User or Operator Response: Check the TYPE field in the defined resource record.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readMaster

EZB3510W Format error 2: *line*

Explanation: The NSDBLOAD program encountered a syntax error in the input file. The line causing the error is displayed and ignored.

System Action: TCPIP continues.

User or Operator Response: The required TYPE field is missing from the resource record.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readMaster

EZB3511W Format error 3: *line*

Explanation: The NSDBLOAD program encountered a syntax error in the input file. The line causing the error is displayed and ignored.

System Action: TCPIP continues.

User or Operator Response: The required TYPE field is missing from the resource record.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readMaster

EZB3512W Obsolete root definition replace with '.'

Explanation: The NSDBLOAD program encountered an obsolete root definition (..) in the input file. The NSDBLOAD program ignores this error.

System Action: TCPIP continues.

User or Operator Response: Change the root definition from '..' to '.'.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3513W Invalid wild card definition

Explanation: The NSDBLOAD program encountered an incorrect wildcard definition in the input file.

System Action: TCPIP continues.

User or Operator Response: Correct the input file. See RFC 1035 for the correct wild card entry.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3514W Mailbox in SOA data field is specified incorrectly.

Explanation: The NSDBLOAD program encountered an incorrectly specified *RNAME* field within an *SOA RR RDATA* field while reading the master data file.

System Action: TCPIP continues.

User or Operator Response: See RFC 1034 for the proper specification of the RNAME field.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: readmaster

EZB3515I Drop index *sql table*

Explanation: NSDBLOAD will preform an SQL drop index command against the table displayed in the message

System Action: NSDBLOAD performs the SQL drop index command and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: dropindex

EZB3516I *SQLcommand*

Explanation: The name server will perform a create index SQL command against the SQL table displayed in the message.

System Action: The NSDBLOAD performs the create index command and continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: createndex

EZB3517I * WARNING: OWNER of SQL table is not NAMESEV. ****

EZB3518I * Ensure view exists on** *owner***.nstable. ****

Explanation: The current name server SQL table is not owned by NAMESRV.

System Action: NSDBLOAD continues.

User or Operator Response: Make sure that the owner specified in the message has view authority for the table.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3519I Select tcurrent

Explanation: NSDBLOAD is selecting a table to receive SQL data.

System Action: NSDBLOAD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3520I PREPARE SQL error: *sqlcode*

Explanation: NSDBLOAD did a PREPARE of the SQL DB2 table, and SQL returned the code that is displayed in the message.

System Action: NSDBLOAD continues.

User or Operator Response: Use the *sqlcode* that is displayed in this message and *DB2 Messages and Codes* to determine the cause of the error and respond as indicated.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3521I Declare SQL error: *sqlcode*

Explanation: NSDBLOAD did a DECLARE of the SQL DB2 table and received the return code that is displayed in the message.

System Action: NSDBLOAD continues.

User or Operator Response: Use the *sqlcode* that is displayed in this message and *DB2 Messages and Codes* to determine the cause of the error and respond as indicated.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3522I Open SQL error: *sqlcode*

Explanation: NSDBLOAD did an OPEN of the SQL DB2 table and received the return code that is displayed in the message.

System Action: NSDBLOAD continues.

User or Operator Response: Use the *sqlcode* that is displayed in this message and *SGL/DS Messages and Codes* to determine the cause of the error and respond as indicated.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3523I Table *ttable* **not defined in nstable.**

EZB3524I using default table *sqltable*

Explanation: NSDBLOAD performed an SQL FETCH command. The table displayed in the message was not defined in the nstable.

System Action: NSDBLOAD uses the default table and continues.

User or Operator Response: Verify that the table specified in the NSDBLOAD command was valid and it has been defined in the DNSTABLE data set.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3525I Using sql table *table*

Explanation: The table displayed is being used by NSDBLOAD.

System Action: NSDBLOAD continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: whichtable

EZB3534I

EZB3535I **NSDBLOAD reads a master dataset, as defined in RFC1034,**

EZB3536I **generates resource records, and inserts them into an SQL**

EZB3537I **table. Multiple name servers can execute simultaneously**

EZB3538I **on the same system assuming they are using separate ports**

EZB3539I **or are executing on different TCPIP service machines.**

EZB3540I **When inserting data into an SQL table owned by NAMESRV the**

EZB3541I **above SQL names are permitted. When inserting into an SQL**

EZB3542I **table NOT owned by namesrv, the SQL tablename must be fully**

EZB3543I **qualified. If a view is defined on nstable, NSDBLOAD will**

EZB3544I **determine from the status field which table is current (0|1)**

EZB3545I **Format: nsdbload [db2-subsystem sqltable input-dataset**

EZB3546I **output-dataset]**

EZB3547I **where: db2name – SQL subsystem name**

EZB3548I **sqltable – the table owner defaults to NAMESRV.**

EZB3549I **Any of the following forms are permitted:**

EZB3550I **cache, namesrv.cache, or namesrv.cache0.**

EZB3551I **input-dataset – master data dataset**

EZB3552I **output-dataset – resource record dataset**

Explanation: NSDBLOAD displays this help text if you specified a question mark as the only command line parameter.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: makelower

EZB3560I *time*: **Do you want to create resource records from a master file (y/n)?**

Explanation: The NSDBLOAD command is prompting you to determine if the resource records should be created from a master file or the program should end now.

System Action: TCPIP continues.

User or Operator Response: Enter Y or N.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: main

EZB3561I *time*: **Do you want to delete all resource records from db(y/n)?**

Explanation: The NSDBLOAD command is prompting you to determine if all resource records should be deleted from the database.

System Action: TCPIP continues.

User or Operator Response: Enter Y or N.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: main

EZB3562I *time*: **Do you want to insert the RR in the sqltable now(y/n)?**

Explanation: The NSDBLOAD command is prompting you to determine if the resource records defined in the input parameters should be inserted into the SQL tables now.

System Action: TCPIP continues.

User or Operator Response: Enter Y or N.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: main

EZB3563I *time*: **Do you want translate all data to lower case?(y/n)**

Explanation: The NSDBLOAD command is prompting you to determine if all input data should be converted to lowercase.

System Action: TCPIP continues.

User or Operator Response: Enter Y or N.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3564I *time*: **Do you want translate all data to upper case?(y/n)**

Explanation: NSDBLOAD is prompting you to determine if all input data should be translated to uppercase.

System Action: TCPIP continues.

User or Operator Response: Enter Y or N.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: insertsql

EZB3565I *time*: **Drop index** *table*

Explanation: NSDBLOAD is issuing an SQL DROP INDEX command for this table.

System Action: TCPIP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name: dropindex

EZB3566I **RDATA Error:** *resource record*

Explanation: The resource record displayed is missing the RDATA field.

System Action: NLDBLOAD increments an error counter and continues.

User or Operator Response: Correct the resource record of the name server SQL table.

System Programmer Response: None.

Source Data Set: NSDBLOAD

Procedure Name:

Chapter 7. EZB3825—EZB4198

NCPROUTE Messages

This section contains NCPROUTE messages.

EZB3825T *name***/udp: unknown service**

Explanation: A UDP port number for the service *name* was not assigned in the tcpip.v3r1.ETC.SERVICES data set.

System Action: NCPROUTE exits.

User or Operator Response: None.

System Programmer Response: Verify that the tcpip.v3r1.ETC.SERVICES data set has two entries in the form:
ncproute
port/udp
router *port*/udp

The entries must start in column 1 and be in lowercase. Verify that the NCPROUTE service port number is the port being used by the NCP clients. The port value defined in the UDPPORT= keyword on the IPOWNER statement in the NCP generation definition must match the NCPROUTE service port. The default is UDP port 580.

The reserved router service port number is 520 and is required for the NCPROUTE transport of RIP packets to NCP clients, which are responsible for broadcasting the packets to other RIP routers. The router service port number cannot be overridden due to NCP restriction.

Also, verify that port 580 has been reserved for NCPROUTE under the PORT statement in the tcpip.v3r1.PROFILE.TCPIP data set.

Source Data Set: NRMAIN

Procedure Name: main

EZB3826I **Port** *port* **assigned to** *name*

Explanation: NCPROUTE will listen for traffic from NCP clients on the specified port *port* assigned to service *name*.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If communications cannot be established with a client, compare NCPROUTE's port number with the value specified in the client's NCP generation definition. The port value defined in the UDPPORT= keyword on the IPOWNER statement in the NCP generation definition must match.

Source Data Set: NRMAIN

Procedure Name: main

EZB3827T **Terminating since clients require the socket**

Explanation: NCPROUTE attempted to open a socket on a well known or user-defined port but the open was not successful, or the socket could not be bound to an internet address and port number. Clients will not be able to communicate with NCPROUTE because a socket is not available.

System Action: NCPROUTE 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 tcperror() library message. Correct the problem as indicated by error. Refer to the *OS/390 C/C++ Run-Time Library Reference* for more information about socket() function errors.

Source Data Set: NRMAIN

Procedure Name: Main

EZB3828T Usage: NCPROUTE *parameters*

Explanation: Incorrect parameters were passed to NCPROUTE.

System Action: NCPROUTE ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that the parameters are correct. The parameters are case-sensitive and must be separated by spaces. See *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRMAIN

Procedure Name: main

EZB3829I Waiting for incoming packets

Explanation: NCPROUTE is waiting for datagrams from NCP clients. Each time that NCPROUTE finishes processing an event, such as an incoming datagram or a timer that expires, NCPROUTE issues this message and waits for the next event. These messages should occur at least once every 30 seconds, but will increase in frequency as the server performs more work.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Check to see if the assigned port number for NCPROUTE matches the NCP clients' generation definitions.

If 6 or more of these messages occur consecutively, NCPROUTE is not receiving any datagrams from NCP clients. Verify that a client has an established session with NCPROUTE, and if one exists, examine the status of the client's interfaces.

Source Data Set: NRMAIN

Procedure Name: main

EZB3830E The main select was interrupted:

Explanation: An error occurred while NCPROUTE was waiting for an event to occur. A more detailed tcperror() library message follows.

System Action: NCPROUTE continues, unless an IUCV error occurred.

User or Operator Response: None.

System Programmer Response: Verify that TCPIP is running.

Source Data Set: NRMAIN

Procedure Name: Main

EZB3831I Send delayed dynamic update

Explanation: A routing update, which had been delayed to prevent packet storms, has been transmitted. This occurs 2–5 seconds after a dynamic update has been issued.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: Main

EZB3832E While receiving a packet from a client:

Explanation: An error occurred while attempting to receive a packet from a client. A more detailed tcperror() library message follows.

System Action: NCPROUTE continues, and the incoming packet is discarded. If an IUCV error occurs, NCPROUTE will end.

User or Operator Response: None.

System Programmer Response: See the next generated error message and correct the error.

Source Data Set: NRMAIN

Procedure Name: process

EZB3833E While receiving a packet from the SNMP agent: *agent*

Explanation: An error occurred while attempting to read a packet from the SNMP agent. A more detailed tcperror() library message follows. If the connection with the Agent has been reset, the SNMP agent will be terminated and incoming SNMP requests will be ignored.

System Action: NCPROUTE continues. The SNMP packet is discarded. If an IUCV error occurs, NCPROUTE will end.

User or Operator Response: None.

System Programmer Response: If the connection with the SNMP agent has been reset, restart the SNMP daemon (OSNMPD).

Source Data Set: NRMAIN

Procedure Name: read_dpi

EZB3834I ******** **

Explanation: Two of these banners enclose a message that may need attention. when viewing the output. The severity of the enclosed message is indicated.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Read the enclosed message and, if necessary, resolve the indicated situation.

Source Data Set: *various*.

Procedure Name: *various*.

EZB3835T Invalid parameter: *parameter*

Explanation: An incorrect parameter was passed from the tcpip.v3r1.SEZAINST(NCPROUT) start proc JCL. The parameter could be passed from the command line parameters or from the default parameter list in the start proc JCL.

System Action: NCPROUTE exits.

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 proc JCL.

Source Data Set: NRMAIN

Procedure Name: main

EZB3836W The SNMP agent has terminated

Explanation: The socket used to communicate with the SNMP agent has been reset. This implies that the SNMP agent has ended. Without this socket, SNMP requests are not processed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If the SNMP agent (OSNMPD) is not running, restart the agent. NCPROUTE will attempt to reestablish communications with the agent after each new SNMP request is received, so no further action is required. If the agent is running, contact the IBM Software Support Center.

Source Data Set: NRMAIN

Procedure Name: read_dpi

EZB3837W SNMP requests will be ignored until it is restarted.

Explanation: Because communication with the SNMP agent is not possible, incoming SNMP requests are ignored until a connection has been established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Start the SNMP agent (OSNMPD). NCPROUTE will attempt to establish communication with the agent for each SNMP packet received until a connection is established.

Source Data Set: NRMAIN

Procedure Name: read_dpi

EZB3838W An SNMP DPI packet arrived from a non-internet machine

Explanation: An SNMP DPI packet was received from a non-internet network. SNMP DPI traffic is only accepted from machines running TCPIP. The incoming packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the machine generating the SNMP DPI packet and have it stopped. Ignore these messages.

Source Data Set: NRMAIN

Procedure Name: read_dpi

EZB3839T A socket could not be created:

Explanation: NCPROUTE could not open a new socket. With the socket unavailable, the NCP clients will not be able to establish communications with NCPROUTE. A more detailed tcperror() library message follows.

System Action: NCPROUTE ends abnormally.

User or Operator Response: None.

System Programmer Response: Verify that TCPIP is active, that another program is not using NCPROUTE's port, that the well-known port has been reserved in the PORT statement of *hlq*.PROFILE.TCPIP data set, and that the user-defined port has been specified in *hlq*.ETC.SERVICES data set. Correct the problem as indicated by the error in the detailed tcperror() library message. Refer to the *OS/390 C/C++ Run-Time Library Reference* for more information about socket() function errors.

Source Data Set: NRMAIN

Procedure Name: getsocket

EZB3840T Broadcasting cannot be enabled on the socket:

Explanation: NCPROUTE cannot enable the socket for broadcasting. NCPROUTE must be able to broadcast over interfaces which support broadcasting in order to communicate with the NCP clients. A more detailed tcperror() library message follows.

System Action: NCPROUTE ends abnormally.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: NRMAIN

Procedure Name: getsocket

EZB3841T The socket bind failed

Explanation: NCPROUTE was unable to associate an internet address and port number to the newly created socket. Another application could be using the port. A more detailed tcperror() library message follows.

System Action: NCPROUTE ends abnormally.

User or Operator Response: None.

System Programmer Response: Use the NETSTAT ALLCONN command to verify that there is no other application using the NCPROUTE port number. Look for a port in the "Local Socket" column which matches NCPROUTE's port in the *hlq*.ETC.SERVICES data set. You should reserve this port for NCPROUTE's exclusive use by adding an entry to the PORT statement in the *hlq*.PROFILE.TCPIP data set.

Source Data Set: NRMAIN

Procedure Name: Getsocket

EZB3842W Hello from existing client *client*

Explanation: An NCP client with a current session has entered a reset state and has started to send Hello packets in an attempt to establish a session with NCPROUTE. This client might have been shut down and restarted. The current session ends and a new one is started.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: Pdu_in

EZB3843W Status from a session-less client *client*

Explanation: A client has transmitted an interface status change PDU without first establishing a session with NCPROUTE. This is a protocol violation. The PDU is discarded. Also, the client is added to a "Client Protocol Violation List" and no further error messages will be issued for this client until it successfully establishes a session. At that point, NCPROUTE will accept the client, and the client's name will be removed from the list.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Restart the client NCP. If the problem persists, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: pdu_in

EZB3844W This datagram is being ignored

Explanation: A datagram was received and some outstanding error prevents NCPROUTE from operating on the packet. The most likely cause is that a protocol error occurred between the client and NCPROUTE. The datagram is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Identify the cause of the error from a previous error message and correct the problem. The client must be reset so that a new session is established.

Source Data Set: NRPDUS

Procedure Name: pdu_in

EZB3845W Transport from session-less client *client*

Explanation: A client NCP has transmitted a Transport PDU without first establishing a session with NCPROUTE. This is a protocol violation. The PDU is discarded. Also, the client is added to a "Client Protocol Violation List" and no further error messages will be issued for this client until it successfully establishes a session. At that point, NCPROUTE will accept the client and the client's name will be removed from the list.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Restart the client NCP. If the problem persists, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: pdu_in

EZB3846W InactList from session-less client *client*

Explanation: A client NCP has transmitted an inactive interface PDU without first establishing a session with NCPROUTE. This is a protocol violation. The PDU is discarded. Also, the client is added to a "Client Protocol Violation List" and no further error messages will be issued for this client until it successfully establishes a session. At that point, NCPROUTE will accept the client and the client's name will be removed from the list.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Restart the client NCP. If the problem persists, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: pdu_in

EZB3847W Protocol Violation: client *client* **sent**

Explanation: A foreign machine sent a packet to the NCPROUTE port that does not contain a valid type field in the packet header. This is probably not an NCP, but another machine on the network. The PDU is discarded. Also, the client is added to a "Client Protocol Violation List" and no further error messages will be issued for this client until it successfully establishes a session. At that point, NCPROUTE will accept the client and the client's name will be removed from the list.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Identify the source of the packet from the displayed internet address and correct the problem. If the machine in question is an NCP, verify that it is loaded with the correct software and reset it. If the machine is an NCP and resetting it does not correct the problem, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: pdu_in

EZB3848W An excessive number of clients (#) have been issued protocol violations. Further warning messages will be suppressed.

Explanation: A large number of clients have committed protocol violations and have not subsequently established sessions with NCPROUTE. No further "Warning" messages are issued. NCPROUTE assumes that the problems have been noticed by this point and that steps are being taken to correct them.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the most recent versions of NCPROUTE and the NCP drivers are installed. Verify that previous protocol errors on individual client NCPs were investigated and corrective actions were taken. Contact the IBM Software Support Center if the protocol violations continue.

Source Data Set: NRPDUS

Procedure Name: bad_client

EZB3850E Status change for unknown interface *interface*

Explanation: NCPROUTE received a status change request PDU from the NCP client for an interface that was unknown. Either NCPROUTE did not completely read in the NCP client's Routing Information Table (RIT), or the RIT was not built correctly during NCP generation. Another possible cause is that the NCP client dynamically added the interface to its tables. Dynamically-added interfaces are not currently supported by NCPROUTE. The Status PDU is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Check the interface configuration messages created in the console log at install time to verify that the interfaces defined in the NCP client's generation match the interfaces processed by NCPROUTE. If a mismatch is found, verify that the RIT is built correctly. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3851I Accepting bad client *client*

Explanation: A client that committed a protocol violation earlier has successfully established a session and any new protocol violations are reported.

System Action: The client is removed the "Client Protocol Violation List". With the client removed from this list, NCPROUTE resumes reporting any new protocol violation errors.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: forgive

EZB3855I NCP_Add out to *client* **Route to** *address1* **via interface** *interface* **to** *address2* **Metric:** *metric***, Type** *type***, Subnetmask** *mask*

Explanation: An ″Add″ PDU is being sent to a client, which causes a route to be added to its IP route tables.

address1 is the destination internet address of the route.

The *interface* is the name given to the interface during NCP generation.

address2 is the internet address of an intermediate router or zero if the destination is directly connected.

The *metric* is the relative cost of using this route as opposed to another route.

The *type* is either Host, Subnet, or Network and indicates the route type being added.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: nr_do_add

EZB3856I Pending delete for interface *interface*

Explanation: NCPROUTE received a status change request from the NCP client to delete the specified interface from its interface tables. The NCP client had deleted the interface from its configuration as a result of dynamic reconfiguration. NCPROUTE puts the interface in pending delete state so that routing outages can be reported to other interfaces and the incorrect addition of routes in routing responses can be prevented. NCPROUTE will continue to remove the specified interface from its tables until either the routes are timed out or a status change request to add a new interface with the same internet address is received.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3857I Adding new interface *interface*

Explanation: NCPROUTE received a status change request from the NCP client to add a new interface to its interface tables. The NCP client had added the interface to its configuration as a result of dynamic reconfiguration. NCPROUTE will manage routes for the new interface.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3858E Unknown status change request

Explanation: NCPROUTE received an invalid status change request from the NCP client. The Status PDU is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3859I Deleting interface *interface*

Explanation: NCPROUTE is deleting the specified interface from its interface tables. The interface was previously in pending delete state so that routing outages could be reported to other interfaces and the incorrect addition of routes in routing responses could be prevented. The deletion occurs when either the routes attached to the interface have timed out or a status change request to add a new interface with the same internet address is received.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3860E The transmission of an 'Add' to client *client* **failed**

Explanation: TCPIP detected that the ″Add″ PDU could not be delivered successfully. The add request, for adding a route to the NCP client's routing table, is not performed. A more detailed tcperror() library message follows.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the error message that follows and proceed according to the recommendations.

Source Data Set: NRPDUS

Procedure Name: nr_do_add

EZB3862I NCP_Add out to *client* **Route to** *address1* **via interface** *interface* **to** *address2* **Metric:** *metric***, Type** *type***, Subnetmask** *mask*

Explanation: A ″Delete″ PDU is being sent to the named client. The *address* is the destination of the route to be deleted, and the *type* is either Host, Subnet or Network and is used by the client to locate the correct route table from which to delete the route.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: nr_do_delete

EZB3863S RIP2 authentication key exceeds maximum allowed or contains unsupported characters

Explanation: The RIP Version 2 authentication key, specified on the RIP2_AUTHENTICATION_KEY entry in the NCPROUTE profile data set, or on the options statement entry in the NCP client's gateways data set for an interface, is invalid. The authentication key may have contained unsupported characters or have exceeded the maximum 16 characters allowed. The authentication key is ignored and no authentication check is performed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the NCPROUTE profile or the NCP client's GATEWAYS data set.

Source Data Set: NRMAIN, NRSTART

Procedure Name: read_profile, ParseOptions

EZB3864E The transmission of a 'Delete' to client *client* failed

Explanation: TCPIP detected that the ″Delete″ PDU could not be delivered to an NCP client. The delete request, for deleting a route from the NCP client's routing table, is not performed. A more detailed tcperror() library message follows.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the error message that follows and proceed according to the recommendations.

Source Data Set: NRPDUS

Procedure Name: nr_do_delete

EZB3865T Out of memory during transport

Explanation: NCPROUTE has used all available storage while attempting to send a ″Transport″ PDU to a client. In this condition, NCPROUTE can no longer communicate with its clients.

System Action: NCPROUTE exits.

User or Operator Response: None.

System Programmer Response: Increase NCPROUTE's region size and restart. Take into consideration that storage requirements are based on the number of clients being served and the number of routes being managed. If the problem still cannot be resolved, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: nr_do_transport

EZB3866E The transmission of a 'Transport' to client *client* failed

Explanation: TCPIP detected that it would be unable to deliver the "Transport" PDU to a client. The ″Transport″ PDU is discarded. A more detailed tcperror() library message follows.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the error message that follows and proceed according to the recommendations.

Source Data Set: NRPDUS

Procedure Name: nr_do_transport

EZB3867I Acknowledge to *client*: Hello Received

Explanation: A client has attempted to establish a session with the server and is being updated on the status of the request. The ″Hello″ PDU was received and is sent immediately to quiet the client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3868I Acknowledge to *client*: RIT Loaded OK

Explanation: A client has attempted to establish a session with the server, and is being updated on the status of the request. The Routing Information Table (RIT) was loaded and appears usable.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3869E Acknowledge to *client*: RIT Load Failed

Explanation: A client has attempted to establish a session with the server and is being updated on the status of the request. An abend occurred during the load of the Routing Information Table (RIT). The session with the server is not established. A previous message explains the cause of the abend.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the abnormal end code in a previous message and correct the error. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3870E Acknowledge to *client*: RIT ID Bad

Explanation: A client has attempted to establish a session with the server and is being updated on the status of the request. An error was detected in the correlation string of the Routing Information Table (RIT). The string does not match the one in the received ″Hello″ PDU. Most likely the NCP client has been reconfigured after a new NCP load and, as a result, the RIT ID was updated. The session with the server is not established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the correct RIT was installed and that another data set was not inadvertently loaded earlier. The RIT should be referenced in the operating system's link list or in the DD:STEPLIB statement of the NCPROUTE catalogued procedure. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3871E Acknowledge to *client*: RIT Bad

Explanation: A client has attempted to establish a session with the server and is being updated on the status of the request. An error was detected in the Routing Information Table (RIT) data. The session with the server is not established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the correct RIT was installed and that another data set was not inadvertently loaded earlier. The RIT should be referenced in the operating system's link list or in the DD:STEPLIB statement of the NCPROUTE catalogued procedure. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3872E Acknowledge to *client*: RIT Not Found

Explanation: A client has attempted to establish a session with the server and is being updated on the status of the request. The Routing Information Table (RIT) could not be found matching the NCP name in the received ″Hello″ PDU. The session with the server is not established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the RIT built during NCP generation has been installed into a partitioned data set and referenced by the DD:STEPLIB statement of NCPROUTE catalogued procedure. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3873E Acknowledge to *client*: Unsupported Ack Type

Explanation: A client has attempted to establish a session with the server, and is being updated on the status of the request. An unknown Acknowledgement type was received. The session with the server is not established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3875W Variable subnetting not supported by client *client*

Explanation: Variable subnetting is not supported by the *client* and the RIP supply/receive control setting may have been set in the NCPROUTE configuration such that RIP Version 2 packets are to be sent or received over interface(s). NCPROUTE will override the control settings to RIP1 for compatibility with the NCP client configuration.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify the NCP software level for variable subnetting support in the NCP client. If RIP Version 2 packets are not to be used, correct the NCPROUTE profile data set or the NCP client's gateways data set to use RIP Version 1 packets.

Source Data Set: various

Procedure Name: various

EZB3876I Hello from new client *client*

Explanation: An NCP client is attempting to establish a session with NCPROUTE. This is done by sending a "Hello" PDU containing the NCP name and correlation string used to verify that the correct Routing Information Table (RIT) is loaded.

Note: The NCP client will issue many ″Hello″ PDUs up to the maximum specified in the NCP gen until it successfully establishes a session with the server. Once the maximum is reached, the NCP client will repeat the cycle again after a 9-minute delay timer has expired. The delay timer is used to prevent NCP alert flooding. The NCP client's issuance of ″Hello″ PDUs can happen even after the server has established a session with the NCP client. For example, the NCP client could have been restarted. An attempt will be made to establish a new session.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3877I RIT dataset name: *ritdsname*

Explanation: The specified Routing Information Table (RIT) data set name, as supplied in the Hello PDU, is loaded and processed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3878I RIT ID: *table_id*

Explanation: The specified ID string is used as a correlation string to verify that the correct Routing Information Table (RIT) has been loaded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None, however, if this ID appears unusual, determine if the correlation string was overridden during NCP load.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3879E *timestamp*

Explanation: An error occurred and a full time stamp is written showing the date and time when the error occurred. While each message is timestamped, date information is omitted, and this message is issued to help those running for long periods of time.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: Recv_hello

EZB3880E Error opening RIT *client1* **for client** *client2*

Explanation: The Routing Information Table (RIT), which was created during NCP generation, could not be opened. *client1* is the name of the member on which the load was attempted, *client2* is the internet address of the client that is attempting to establish a session with NCPROUTE. NCPROUTE does not establish the session.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the abend code and reason code in the following EZB3881E message for the specific cause of the error. In the event of an 804 abend, verify that the RIT exists and is accessible to NCPROUTE.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3881E Abend code: *abend*, **Reason:** *reason*

Explanation: An abend occurred during an attempt to load the Routing Information Table (RIT) into memory with the LOAD macro. The abend and reason code returned are displayed. The session with the server is not established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: See *OS/390 MVS System Codes* for an explanation of the abend and reason codes, and further instructions.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3882E A session will not be established

Explanation: An error occurred during an attempt to establish a session with a client, and the session cannot be established at this time.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the errors described in previous messages.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3883E Client: *client*

Explanation: An error occurred during a transaction with the client. The client is the internet address of the client. This message is used to identify the failing client, and is followed by additional messages.

System Action: NCPROUTE issues additional messages.

User or Operator Response: None.

System Programmer Response: Examine the messages that follow for a description of the error.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3884E The RIT ID field does not match the one in the Hello. RIT *idfield*, **Hello:** *correlfield*

Explanation: The ID field in the Routing Information Table (RIT) does not match the correlation field passed from the client NCP in the hello datagram. The mismatched id and correlation fields are displayed. This indicates that the client is using a generation that is different from the one that built the RIT. A session is not established.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Generally, the RIT used by NCPROUTE should be the RIT that was generated during the generation of the current NCP load. In this case, an older RIT with the same name is probably being loaded. Locate and remove the old RIT. In some cases, multiple machines might want to own a single NCP and use the same NCP generation, but would require a unique RIT to run properly. See the NCP documentation for instructions about how to set the correlation string in your generations and during NCP load.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3885I Input parameters: *parms*

Explanation: This message lists the string of input parameters passed to NCPROUTE from tcpip.v3r1.SEZAINST(NCPROUT) start proc JCL. The parameters could either be passed from the command line parameters or from the defined parameter list in the start proc JCL. If no parameters are specified, '*None*' will appear in the string.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: main

EZB3886E An error occurred while loading the RIT for client *client***.**

Explanation: An internal error occurred. As a result of the error, the information in the Routing Information Table (RIT) is not available but an error condition was not indicated on return from the load. The new session ends.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3887E An internal error occurred, terminating session.

Explanation: Because a session was established, but is unusable, the new session ends and this message is displayed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3888E An error occurred during interface initialization.

Explanation: An error occurred during processing of the Routing Information Table (RIT) interface list. Previous messages should describe the nature of the error. The new session ends.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine previous messages and follow recommendations for the error that occurred.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3889E The session will be terminated

Explanation: An error occurred that makes one of NCPROUTE's sessions unusable. The session ends, and the client NCP goes into a reset state within 3 minutes. At that point, the NCP client attempts to reestablish a session.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine previous messages to determine what error occurred and correct the problem. If the error takes a while to correct, renaming the Routing Information Table (RIT) causes future attempts at establishing a session to be unsuccessful.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB3890I Recv: status from *client*

Explanation: A client's interface has changed state and the client has reported this by an interface status change PDU.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3891I Interface *ip_addr* **is now** *status - interface_name*

Explanation: The client's interface name having an internet address of *ip_addr* has changed state and is currently set to either "up" or "down".

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If the status change is not expected, examine the client NCP to determine the reason for the change. See the NCP documentation for more information.

Source Data Set: NRPDUS

Procedure Name: recv_status

EZB3894I Transport from *client***:** *count* **bytes of RIP data.**

Explanation: Client *client* has received a packet addressed to UDP port 520 and has forwarded this packet to NCPROUTE in a Transport PDU.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_transport

EZB3895I Transport from *client***:** *count* **bytes of SNMP data.**

Explanation: The *client* has received a packet addressed to UDP port 161 and has forwarded this to NCPROUTE in a transport PDU.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_transport

EZB3896E Transport for unsupported port *port number* **bytes received and discarded**

Explanation: The client NCP has sent NCPROUTE a Transport PDU containing a packet that was addressed to a port that NCPROUTE does not support. The transport is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: recv_transport

EZB3897I NCPROUTE Server started

Explanation: NCPROUTE has completed initialization and is waiting for incoming packets from NCP client(s) to initiate sessions.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: main

EZB3898I Recv: Inactive Interface List from *client number* **interface(s) found:**

Explanation: After a session is established, the client sends NCPROUTE a list containing the number of interfaces that are currently down. All other interfaces (listed in the Routing Information Table (RIT)) are assumed to be active. Following this message, the IP addresses and names of the inactive interfaces are displayed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_inactlist

EZB3899I *ipaddr - interface*

Explanation: The IP address and name of the inactive interface are displayed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_inactlist

EZB3900W Unable to open NCPROUTE profile *profile*

Explanation: A profile data set was not specified or could not be opened. This message indicates which data set the open was attempted on.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the profile data set is defined in //DD:NCPRPROF of the NCPROUTE start proc JCL and is accessible by NCPROUTE. If the data set is sequential, ensure that the FREE=CLOSE

parameter is specified. Do not specify this parameter if the data set is partitioned.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB3901W 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 by the NCP client. A supernet route is one where its subnet mask is less than the route's network class mask.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify the NCP software level for supernetting support in the NCP client.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3902W 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 make sure that a network user does not pretend to be a router to change the routing table of nearby routers (such as the client NCP). The address is not considered as a valid destination address for a route, and the route entry is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Identify the machine or user that generated the packet in question and correct the problem.

Source Data Set: NRAF

Procedure Name: inet_checkhost

EZB3903W Invalid internet address

Explanation: An internet address in an incoming route is determined to not be a member of any defined internet address class. The route is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the router that originated this packet.

Source Data Set: NRAF

Procedure Name: inet_checkhost

EZB3904W A must-be-zero field is non-zero

Explanation: An internet address in an incoming route contains a nonzero value in a field that must be 0. The incoming route is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the router that originated this packet.

Source Data Set: NRAF

Procedure Name: inet_checkhost

EZB3905S function *function* **client** *client* **unknown**

Explanation: The function was unable to obtain a list of interfaces for the client with internet address *client*. If *client* is an actual client of NCPROUTE, the interface list has been lost, otherwise the function is using an incorrect address for its client. NCPROUTE indicates that the requested interface does not exist.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRIF

Procedure Name: ifwithaddr

EZB3906S NULL interface list for client *client*

Explanation: A list of interfaces for client *client* exists, but is marked empty. Dynamic routing requires at least one interface. The requested interface is indicated as nonexistent.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the client NCP generation had at least one interface that did not have the RIPMGD=NO keyword coded. If none are found, correct the configuration and regenerate the client. Verify that EZB3956I messages are issued which correspond with the Routing Information Protocol (RIP)-managed interfaces in the generation. If none are found, attempt to reload the Routing Information Table (RIT). Contact the IBM Software Support Center if the problem persists.

Source Data Set: NRIF

Procedure Name: ifwithaddr

EZB3908I Modify command is set for all clients

Explanation: NCPROUTE is configured to process MODIFY commands for all clients rather than for a specific client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: do_modify

EZB3910I Modify command is set for client *client*

Explanation: NCPROUTE is configured to process MODIFY commands for a targeted client rather than for all clients.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: do_modify

EZB3911S Function *function* **address family out of range. The address is** *address***.**

Explanation: One of NCPROUTE's route entries has an unsupported address family. NCPROUTE cannot determine the network based on the address, therefore it cannot determine an interface that serves the logical network. The requested interface is indicated as nonexistent.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Noninternet addresses are not supported in this version of NCPROUTE. Either ignore the message, or have the router that originated this route stop sending noninternet routes to the client NCP. Look back through the output to find the last ADD or CHANGE for this destination to obtain the internet address of the router.

Source Data Set: NRIF

Procedure Name: ifwithnet

EZB3912I ifwithnet: compare with *interface*

Explanation: The route's network address is being compared with one of NCPROUTE's interface entries for a network number match.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRIF

Procedure Name: if_ifwithnet

EZB3913I ifwithnet: remote interface ignored

Explanation: One of NCPROUTE's interface entries is for a remote interface, which is ignored during the search for a network number match.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRIF

Procedure Name: if_ifwithnet

EZB3914S ifwithnet: interface has bad address family

Explanation: One of NCPROUTE's interface entries has an incorrect address family. NCPROUTE cannot determine the network based on the address. Therefore, it cannot find an interface that serves the logical network.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Either ignore the message or correct the NCP client's interface that contains the incorrect address family. If the interface cannot be corrected, contact your IBM Software Support Center.

Source Data Set: NRIF

Procedure Name: if_ifwithnet

EZB3915I netmatch *ipaddr1* **and** *ipaddr2*

Explanation: A network number match was found for the route's network address *ipaddr1* with one of NCPROUTE's interface entries having a network address of *ipaddr2*.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRIF

Procedure Name: if_ifwithnet

EZB3916I Blocking route for *destination*

Explanation: From a packet received over a particular interface, the route for *destination* is being blocked from being added to NCPROUTE's routing table. This is the result of a RIP I/O filter option specified in the client's *hlq*.ETC.GATEWAYS data set. See *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3917I NCPROUTE's internal *type* **table for client** *client***:**

Explanation: NCPROUTE's internal IP routing or interface table is displayed for diagnosis.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES, NRIF

Procedure Name: dsp_rrtables, dsp_iftables

EZB3918I Modify parameters processed; see SYSPRINT or SYSERR output for results

Explanation: The modify command with specified parameters has been processed. For results, see the SYSPRINT or SYSERR output.

System Action: NCPRoute continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: do_modify

EZB3921I Tracing debug packets *action timestamp*

Explanation: Debug packets tracing is enabled. The packets are displayed in data format.

System Action: NCPRoute continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: modifydebuglevel

EZB3922S inet_makeaddr: no session with *client*

Explanation: The function inet_makeaddr was unable to obtain a list of interfaces for the client with internet address *client*. If *client* is an actual client of NCPROUTE, the interface list has been lost; otherwise, the function is using an incorrect address for its client. A NULL internet address is created.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRINET

Procedure Name: inet_makeaddr

EZB3923S inet_netof: no session with *client*

Explanation: The function inet_netof was unable to obtain a list of interfaces for the client with internet address *client*. If *client* is an actual client of NCPROUTE, the interface list has been lost; otherwise, the function is using an incorrect address for its client. Because a subnetmask cannot be located, a network number is returned.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRINET

Procedure Name: inet_netof

EZB3924I inet_netof: ignoring REMOTE interface

Explanation: One of NCPROUTE's interface entries is for a remote interface, which is ignored during the calculation for a (sub)network number from the interface's internet address.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINET

Procedure Name: inet_netof

EZB3925S Inet_lnaof: no session with *client*

Explanation: The function inet_lnaof was unable to obtain a list of interfaces for the client with internet address *client*. If *client* is an actual client of NCPROUTE, the interface list has been lost; otherwise, the function is using an incorrect address for its client. Because subnetmask information is not available, the host part of the internet address is returned.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center. This might contain a subnet number if the interface is subnetted.

Source Data Set: NRINET

Procedure Name: inet_lnaof

EZB3926S inet_rtflags: no session with *client*

Explanation: The function inet_rtflags was unable to obtain a list of interfaces for the client with internet address *client*. If *client* is an actual client of NCPROUTE, the interface list has been lost; otherwise, the function is using an incorrect address for its client. A determination is made to indicate either a network address or a host address assuming no subnetting has been done.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRINET

Procedure Name: inet_rtflags

EZB3927E Packet from unsupported address family (*family*), cmd (*command*)

Explanation: A packet was received from a non-internet network. RIP traffic is only accepted from machines running IP. The incoming packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the machine generating the RIP packet and have it stopped. These messages can also be ignored.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3928E RIP version 0 packet received from *ipaddr*

Explanation: A Routing Information Protocol (RIP) Version 0 packet was received from the specified address. This RIP version is obsolete. The incoming packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the router that is sending the Version 0 packets.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3929I Request: output routines removed

Explanation: NCPROUTE received a Routing Information Protocol (RIP) request packet for a specific set of routes rather than for a set of all routes. (A request for all routes is indicated by an address family of zero and infinite metric of 16.) Since NCPROUTE does not support handling requests for a specific set of routes, the output routines in this case have been removed to prevent a response.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3930W Trace command from unknown router *ipaddr*

Explanation: A trace packet was received from a router that is either not directly connected to any of the client's Routing Information Protocol (RIP) managed interfaces, or is directly connected to an interface that is not capable of supporting RIP traffic. RIP requires that an interface be capable of supporting link-level broadcast traffic, be a point-to-point interface (such as NCST sessions), or have an active gateway defined. The incoming packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Determine where the packet originated from and correct the problem. Bridges can allow routers on logically disconnected networks to talk with each other even though the routeres have no logical connection to each other's networks.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3931W Response from non-router *ipaddr*

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 issues RIP routing updates. The RIP response is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Reconfigure the router to use a correct port number or locate the application that is generating the updates and correct the problem.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3932W Invalid packet from passive interface *interface*

Explanation: A Routing Information Protocol (RIP) response packet was received from a passive interface. Passive interfaces receive routing updates from the client, but cannot produce routing updates. The packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the router that is producing the bogus packet and correct the problem.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3933W Packet from unknown router *ipaddress* **(***reason***)**

Explanation: A Routing Information Protocol (RIP) response was received from a router which is not directly connected via a broadcast network, a point-to-point (NCST) network, or an active gateway as defined in a GATEWAYS PDS member. The description of the reason for this warning is one of the following:

Reason Explanation

Interface in strange state
The network does not support broadcast or point-to-point transmission.

Iflookup failed
Not directly connected.

Link is PASSIVE!
Cannot update.

The packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the router that produced the packet and correct the problem.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3937W New route is in unsupported address family. client = *client***, route from** *ipaddr***, new family =** *family*

Explanation: An incoming router from another router is in an address family that is not supported by NCPROUTE. Currently, only internet addresses are supported. The client is the internet address of the client NCP, the *ip addr* is the internet address of the router that originated the route that is not valid, and family is the address family that is not supported by NCPROUTE. The route that is not valid is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the router that produced the route and correct the problem.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3939E Illegal address *hostaddr* **in route from** *ipaddr*

Explanation: An internet address that is not valid was received in an update from router *ipaddr* A previous message indicates the nature of the problem with the address. The route that is not valid is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the router that originated the route and correct the problem.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3940W Bad metric (*metric***) in route to** *destination* **from router** *ipaddr*

Explanation: A route was received from *ipaddr* that contained a metric that was not in the range 1 - 16. The route is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the router ip address and correct the problem.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3941I Adjusting large metric (*metric***) to infinity**

Explanation: From the routing updates, a route contained a metric exceeding the maximum supported metric of 16. The metric is changed to infinite metric of 16, to indicate that the destination route is network unreachable.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3943I Send dynamic update

Explanation: Changes have occurred and an update has not been sent recently. Enough time has passed since the last update was sent so that it is safe to transmit again without risking an update storm. Also, no dynamic update was pending; otherwise, NCPROUTE would wait until that update occurred before sending the update.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3944I Delay dynamic update

Explanation: Changes have occurred in the network topology, but an update was recently made to adjacent routers. When the last update was made, a random delay time, 2-5 seconds from that point, was determined. This update is scheduled to occur at that time.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3945I Inhibit dyanamic update for *seconds* usec

Explanation: A dynamic update has just been sent. Another dynamic update is prevented from occuring for the number of microseconds indicated in the message. A random time is chosen, 2-5 seconds from this time, and if another update is needed later, it will be delayed until this random time has passed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB3946S toall (STUB) -- this shouldn't be used anymore

Explanation: The routine (toall) called by NCPROUTE is no longer supported.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NROUTPUT

Procedure Name: toall

EZB3947S toall: client *client* unknown

Explanation: Information about the client is not available, but NCPROUTE is attempting to send output over all of the client's interfaces. This is an internal error. No output is sent to the client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NROUTPUT

Procedure Name: toall_ifs

EZB3948I Interface *interface* not up

Explanation: The specified interface is detected to be inactive. No route will be added unless the interface is (re)activated.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: addrouteforif

EZB3949I 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 client's *hlq*.ETC.GATEWAYS data set. See *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NROUTPUT

Procedure Name: toall

EZB3950I toall: requested to skip interface *interface*

Explanation: The interface *interface* is skipped because the interface has already received notification of a routing change. If a broadcast of the routing table has not been sent recently and a routing change has occurred, a dynamic routing update will be broadcasted to other interfaces to inform adjacent routers of the routing change.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NROUTPUT

Procedure Name: toall

EZB3951I client *client***: supply** *destaddress* **->** *port* **via** *interfacename*

Explanation: NCPROUTE is directing the client to transmit a Routing Information Protocol (RIP) datagram to the destination address. This can be either a broadcast address or a host address. If the port is zero, the RouteD port will be used from *hlq*.ETC.SERVICES; otherwise, the datagram will be transmitted to the specified port. NCPROUTE will ask the client to use the specified interface name when sending the datagram.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NROUTPUT

Procedure Name: supply

EZB3953S NULL base

Explanation: NCPROUTE was unable to locate a base hash table for the client. Every active client should have this table, so its absence indicates an internal error. No output occurs.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NROUTPUT

Procedure Name: supply

EZB3954W Invalid metric, changing to one

Explanation: In the line entry for the NCP client's *hlq.*.ETC.GATEWAYS data set, the metric has an incorrect value. The metric is changed to 1.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the NCP client's *hlq.*.ETC.GATEWAYS data set.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3952E Unknown interface name (*interface*) and/or address (*ip_addr*)

Explanation: In the line entry for the client's *hlq*.ETC.GATEWAYS data set, the options definition contains interface information that is not in NCPROUTE's interface tables. Most likely, the interface name is misspelled or the interface's internet address is specified incorrectly. Although other options may be processed normally, the invalid option is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq*.ETC.GATEWAYS data set. See *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3955S Function *function*: out of memory

Explanation: NCPROUTE was unable to allocate memory because no more storage is available in the region. The following describe functional errors:

Function	**Error Description**
ifinit	No storage available to add a client interface entry.
nradd	No storage available to add a route table entry. Each route table entry requires 64 bytes.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Increase the region size in the startup procedure.

Source Data Set: various

Procedure Name: various

EZB3956I Processing interface *interface*

Explanation: The indicated interface was found in the Routing Information Table (RIT) for the new client and is being added to NCPROUTE's tables.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: ifinit

EZB3957E Modify command ignored, invalid parm(s): *parms*

Explanation: Invalid parameter(s) were passed to NCPROUTE from a MODIFY command.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the parameter(s) from the MODIFY command.

Source Data Set: NRMAIN

Procedure Name:

EZB3958E Modify command ignored, client 'c=' not specified

Explanation: The parameters specified in the MODIFY command requires the target client specification.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Specify the target client in the ″c=″ parameter of the MODIFY command.

Source Data Set: NRMAIN

Procedure Name: parse_parms

EZB3959I point-to-point interface, using *addrtype*

Explanation: The new interface is point-to-point, and the destination address type, *addrtype*, can either be dstaddr or broadaddr. The address type is determined by the interface definition in the NCP generation. The destination address may be coded such that the network or subnetwork directed broadcast address is used or that the unicast or host address representing the other end of the point-to-point link is used. This address is used for sending 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: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: addrouteforif

EZB3960I This interface is not point-to-point dstaddr

Explanation: The new interface is not point-to-point. A route is being added based on the interface definition, and the network or subnetwork route will be used as the route destination.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: addrouteforif

EZB3961I not an internal interface

Explanation: The new interface does not appear to be associated with a real device, this interface is most likely a pseudo-interface created because of an external route in the GATEWAYS member for this client. No route is created for this interface.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: addrouteforif

EZB3962I Adding *type* route for interface

Explanation: The route using the network, subnetwork or destination type is being added to the interface in NCPROUTE's routing table.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: addrouteforif

EZB3963I Re-installing interface *interface*

Explanation: The previously deleted interface is being re-installed since traffic has been detected over this interface.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: addrouteforif

EZB3965I Route *seq*: *dest* via *gateway* metric *metric*, supnetmask *mask*

Explanation: A route is being added based on an IPROUTE statement coded during NCP generation. The *seq* is the position in the route table, *dest* is the destination IP address, *gateway* is the nexthop for the route, and *metric* is the cost of using this route, and *mask* is the subnet mask for the route.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: bld_rtbl

EZB3967W Unable to open a GATEWAYS dataset for client *client*. Attempt to open *dataset*

Explanation: A gateways data set was not specified, or could not be opened for the specified client. This message indicates which data set the open was attempted on.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If a gateways data set is being used, verify that the PDS member name is coded correctly in the *hlq*.PROFILE.TCPIP data set. Verify that the gateways member has the same name as the client NCP name. Verify that NCPROUTE has access to this data set member. Verify that NCPROUTE attempts to open the correct data set member. Correct the PROFILE and rename the PDS member, if needed. Contact the IBM Software Support Center if the problem persists.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3968I Start of GATEWAYS processing:

Explanation: The gateways data set member is about to be processed. Messages about data set processing might follow.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3969E *statement*

Explanation: The statement shown is in error. A message describing the error will follow.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3970E Invalid gateway address "*gateway*"

Explanation: This statement's gateway is neither a resolvable name nor a valid dot-notation internet address. The statement is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3971E Zero metrics not allowed, changing to one

Explanation: This statement has a 0 metric, which is not valid, Metrics must be between 1 and 15. The metric is changed to 1.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the statement in the gateway member.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3972E Gateway type "passive" not valid for active gateway

Explanation: An active gateway entry is qualified as a passive route. Active is the only valid route type for this definition. The statement is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the statement in the gateway member.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3973I Opening GATEWAYS dataset for client *client. dataset*

Explanation: The gateways data set is being opened for the specified client. Entries in the data set are read in for input.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3974E First two keywords must be 'active' for active gateway.

Explanation: A GATEWAYS entry for the gateway definition contains an active route type, but the first two keywords are not defined as active. These keywords are requered for an active gateway definition. If this entry is not for an active gateway, correct the route type. Active gateway entries identify only a router and have no destination information. The GATEWAYS entry is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the GATEWAYS entry.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3975E Invalid gateway type *type*

Explanation: This statement does not end with a valid route type, either “active”, “passive” or “external”. The statement is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the statement in the gateways member.

Source Data Set: NRSTART

Procedure Name: gateways

EZB3976S nr_exact_find: no session with *client*

Explanation: A route to a destination from a nonexistent client is being requested. This is an internal error. The route is reported as not existing.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nr_exact_find

EZB3977S nr_exact_find: no network hash table for client *client*

Explanation: Client *client* does not appear to have a required network hash table. This is an internal error.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nr_exact_find

EZB3978S nr_kernel_find: no host hash table for client *client*

Explanation: Client *client* appears to be missing a required host hash table. This is an internal error.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nr_kernel_find

EZB3979S nr_kernel_find: no network hash table for client *client*

Explanation: The session with *client* has been improperly initialized. A required network hash table is missing.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nr_kernel_find

EZB3980E nradd: invalid address family

Explanation: An attempt is being made to add a route which is to a destination in an unsupported address family. Currently, NCPROUTE only supports routes to Internet addresses. The add is not performed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Identify the machine that generated the route and correct the problem.

Source Data Set: NRTABLES

Procedure Name: nradd

EZB3981S nradd: no host hash table for client *client*

Explanation: A host route is being added, but the client's host hash table cannot be located. This is an internal error. The route is not added.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Call the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nradd

EZB3982S nradd: no network hash table for client *client*

Explanation: A network or subnetwork route is being added, but the required network hash table for *client* cannot be located. This is an internal error. The route is not added.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Call the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nradd

EZB3983E Modify command ignored, *type* **trace levels exceeded**

Explanation: An incorrect number of trace levels (-t's) were passed from a MODIFY command.

System Action: NCPROUTE 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: NRMAIN

Procedure Name:

EZB3984E error adding route to *host/net destination* **through** *gateway*

Explanation: The NCP client failed to add a route.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine previous error messages to determine the nature of nr_do_add's error.

Source Data Set: NRTABLES

Procedure Name: nradd

EZB3989I *changing/deleting* **route to interface** *interface* **(timed out?)**

Explanation: Either traffic to a local interface is being routed through a remote gateway, or the metric on the interface has increased to infinity. Usually this indicates that the interface has timed out and is considered down.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If the interface is being assigned a new gateway, examine the contents of the last packet sent from the gateway. The gateway's address is found in an upcoming message. Look for a display of the local interface from the remote gateway. Correct the remote gateway if needed. If the interface is timing out, verify that no transmissions have been received over this interface for the last 3 minutes. If no transmissions are found, this is correct behavior for NCPROUTE, and the physical lines or remote gateways should be examined to determine the cause of the problem.

Source Data Set: NRTABLES

Procedure Name: nrchange

EZB3993E could not delete route to *type destination* **via** *router*

Explanation: The route to *destination* is being deleted (possibly because the route is changing, in which case an add will follow) and nr_do_delete returned an error. *type* is one of Host, Network, Subnet, or (bad_type). (bad_type) indicates an internal error and should be reported. Additional errors might follow because of the unsuccessful attempt to remove the route. These should be ignored until this problem is resolved. The route remains.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine previous messages to determine the error that occurred in nr_do_delete, contact the IBM Software Support Center with this information.

Source Data Set: NRTABLES

Procedure Name: nrchange

EZB3994E Error adding route

Explanation: An error occurred while attempting to update a route. The route is not readded and the net effect is that the route is deleted.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine previous messages to determine the error that occurred in nr_do_add and contact the IBM Software Support Center with this information.

Source Data Set: NRTABLES

Procedure Name: nrchange

EZB3996I deleting route to interface *interface*? (timed out?)

Explanation: A route to the indicated interface is being deleted and the metric is less than infinity.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: nrdelete

EZB3997E Error deleting route

Explanation: The NCP clilent failed to add a route. Additional errors might follow this one. Because the route was not deleted, these should be ignored until this problem is resolved.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine previous error messages to determine the error that occurred in nr_do_delete and contact the IBM Software Support Center with this information.

Source Data Set: NRTABLES

Procedure Name: nrdelete

EZB3998S nr_client_init: no session established with *client*

Explanation: A session with the indicated client does not appear to have been initialized properly.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nr_client_init

EZB3999I Establishing session with client *client*

Explanation: A session is being established with a NCP client. This is required before PDUs can be processed from the client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: nr_establish_session

EZB4000I Terminating session with client *client*

Explanation: The session with the indicated client ends. This is an orderly shutdown, and occurs in the following cases: upon receipt of a hello from an active client, if the load of the Routing Information Table (RIT) is unsuccessful, or if an error occurs when processing the RIT. Other messages occur earlier that explain which case caused the session to be brought down.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None required. If the session is brought down in response to an error, examine previous messages to determine the nature of the error, and correct the problem.

Source Data Set: NRTABLES

Procedure Name: nr_terminate_session

EZB4001S Terminate Session: no session with client *client*

Explanation: The NCP client failed to add a route. unacceptable client handle. The client cannot be located in the session table. No session ends.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: nr_terminate_session

EZB4002I Chain *chain_number*

Explanation: The session table is being displayed. Following this message is the contents of chain *chain_number*.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_sessions

EZB4003I Hosts:

Explanation: The session table is being displayed. Following this message is the contents of the host route table for the current session.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_sessions

EZB4004I Networks:

Explanation: The session table is being displayed. Following this message is the contents of the network route table for the current session.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_sessions

EZB4005I No Entries

Explanation: This particular route table is empty.

System Action: NCPROUTE continues

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_rtbl

EZB4006I Subchain *subchain_number*

Explanation: A routing table is being displayed. The following messages contain the contents of the routing table chain *subchain_number*.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_rtbl

EZB4007I Entry empty

Explanation: The route table subchain currently being displayed is empty.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_rtbl

EZB4008I Entry: *entry*

Explanation: The route table subchain currently being displayed contains an entry.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: dsp_rtbl

EZB4009I client *client***:** *timer_value* **minute timer expired for route to** *destination*

Explanation: No Routing Information Protocol (RIP) packets have been received from the NCP client to the destination *destination* in the last *timer_value* minutes. It is assumed that the destination route is no longer active. Depending upon the *timer_value*, one of the following actions is taken:

Value Action

3 The route will have its metric changed to infinity for the next 2 minutes. The metric change is necessary to alert adjacent routers that the route to this destination is unreachable. If NCPROUTE receives any RIP packets for the NCP client from the destination router during the 2 minute time interval, NCPROUTE will restore the route by changing the metric to a valid one based upon the received RIP packet. The route will be deleted from the NCP client's routing table.

5 The route will be deleted from NCPROUTE's routing table for the NCP client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: These actions require the following responses:

Value	Action
3	If the route was broadcasted for a while, and then suddenly stopped, look for an adapter problem, or a problem with the physical line.
5	Examine the NCPROUTE 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 the NCPROUTE's request for full route tables for the NCP client, then the problem may be with the way RIP packets are broadcasted.

In these cases, determine if NCPROUTE is receiving the transport PDUs containing the RIP packets from the NCP client by obtaining a MORETRACE IPUP trace and look for ″discarding broadcast″ packet messages. This trace can confirm the lack of traffic on the interface used for the transport PDUs forwarded by the NCP client. If this is not the case, then determine if NCPROUTE is receiving the RIP packets over the NCP client's routing interface by obtain- ing a NCP line trace. See the NCP documentation for instructions on obtaining a line trace.

Source Data Set: NRTIMER

Procedure Name: client_timer

EZB4010I client *client:* **30 second timer expired (broadcast)**

Explanation: Every 30 seconds, a timer expires that indicates that a client NCP must broadcast its routing tables to adjacent routers. NCPROUTE will build Routing Information Protocol (RIP) response packets for each of the client's interfaces and send them to the client in a transport PDU. The client transmit them on the specified interface.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTIMER

Procedure Name: client_timer

EZB4011E Invalid subnetwork mask *mask*

Explanation: In the line entry for the NCP client's gateways data set, the gateway definition has a subnetmask that is not valid. The subnetmask must be a resolvable subnetmask name, or a bit mask in dotted-decimal notation. The gateway entry is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the NCP client's gateways data set.

Source Data Set: NRSTART

Procedure Name: ParseOptions

EZB4012E Trace buffers not initialized for interface *interface* **on client** *client*

Explanation: One or both trace buffers could not be obtained for the specified interface during initialization. This is because of a lack of free storage within the region. Minimal tracing is performed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Increase the region size for NCPROUTE and restart NCPROUTE.

Source Data Set: NRTRACE

Procedure Name: traceinit

EZB4013I Tracing *action* **for client** *client*

Explanation: Tracing is enabled or disabled for a client NCP.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: various

EZB4014W 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 NCPROUTE profile data set, or specified on the options statement entry for a client for all interfaces or for an interface, contains an incorrect value. In case of incorrect values, NCPROUTE will default the RIP supply control to 'RIP1' and receive control to 'ANY' for the NCP client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the NCPROUTE profile or the NCP client's gateways data set by specifying a supported supply or receive control value.

Source Data Set: NRMAIN, NRSTART

Procedure Name: read_profile, ParseOptions

EZB4015I Client tracing actions started

Explanation: The current tracing level has been advanced to the ″actions″ level, which causes messages to be issued for actions, such as adding, changing, or deleting routes. Additional messages for actions, such as waiting for incoming packets and dynamic updates, are also issued.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: trace_start

EZB4016I Client tracing packets started

Explanation: The current tracing level is advanced to the ″packets″ level, which displays the types of packets sent and received in addition to the output displayed at the ″actions″ level.

System Action: NCPROUTE continues. Correct the *hlq*.NSMAIN.DATA data set to include a server name for the indicated address, or enter the NSLOOKUP command, using the *server_name* and *server_address* parameters to

specify the default Domain Name Server. For more information about the *hlq*.NSMAIN.DATA data set, see the *OS/390 IBM Communications Server: IP Configuration Reference*.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: trace_start

EZB4017I Client tracing history started

Explanation: The current tracing level is advanced to the ″history″ level, which displays history tracing data for each line in addition to output displayed at 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: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: trace_start

EZB4018I Client tracing packet contents started

Explanation: The current tracing level is advanced to the ″packet contents″ level, which displays the contents of packets sent or received in addition to output displayed at lower tracing levels. Additional messages, such as requests for full routing tables and unknown address family in routing information, are also issued.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: trace_start

EZB4029I *timestamp***:**

Explanation: A full time stamp is issued showing the date and time so that traces that exceed one calendar day can be interpreted correctly.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: trace

EZB4030I *action* **destination** *destination* **router** *router***, metric** *metric***, flags** *flags*

Explanation: The route to the destination is being added, deleted, or changed depending on the action. The following action values are allowed:

Value Explanation

ADD The route to the destination is being added through the router at the specified metric.

CHANGE FROM
The route to the destination is being changed and the current values are displayed.

CHANGE TO
The route to the destination is being changed and the new values are displayed.

DELETE
The route to the destination is being deleted.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: trace

EZB4032E Gateway address '*address*' not an routing interface in network

Explanation: The indicated passive gateway, defined in the NCP client's *hlq.*.ETC.GATEWAYS data set, referenced an unknown routing interface based upon the gateway address. The passive route definition is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*.ETC.GATEWAYS data set. Verify that the gateway address is correct and a valid routing interface is defined. See the *OS/390 IBM Communications Server: IP Configuration Reference* for more information about NCP Host Interface Definition.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4036I CHANGE metric destination *destination***, router** *router***, from** *old metric* **to** *new metric*

Explanation: The metric for the route to the destination is being changed from the old metric to the new metric. This is always based on a Routing Information Protocol (RIP) packet from the router:, which updates a previous route through the router.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: tracenewmetric

EZB4038I * Packet history for interface** *interface* *******

Explanation: Tracing is set at the history level, and the history trace data for the inactivated interface *interface* is displayed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumpif

EZB4039I * End packet history *****

Explanation: Tracing is set at the history level, and this message ends the history trace data for the deactivated interface.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumpif

EZB4043I *direction*: **no packets**

Explanation: Either the input or output trace buffer, depending on the direction value, is empty.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumptrace

EZB4044I *direction* **trace:**

Explanation: Tracing is currently at the packet history level. Either the input or output trace buffer is about to be displayed depending on the direction value.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumptrace

EZB4045I *RIPcmd direction router* **->** *port timestam p*

Explanation: A Routing Information Protocol (RIP) datagram has been received, or is about to be sent out of NCPROUTE depending on the direction value. A value of “to” indicates an outbound datagram, while a value of “from” indicates an inbound datagram. The type of the datagram is indicated by *RIPcmd*, and can be either RESPONSE or REQUEST. Responses are displays of routes from one router to another, requests are requests for individual routes or full tables. The value of *port* is either the port number which the packet came in on, the value 0, which indicates that the datagram will be sent to the default *RouteD* port as described in ETC.SERVICES, or in the case of outbound datagrams with a nonzero value the port which the datagram will be sent to.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumppacket

EZB4046S **Bad cmd** *hex direction router* **->** *port timestamp* **size=size cp=directino packet=addr time=timestamp**

Explanation: A malformed packet has been encountered during trace. The source of the packet is either the router, if the value of *direction* is “from”, or NCPROUTE if the value of *direction* is “to”.

System Action: NCPROUTE continues

User or Operator Response: None.

System Programmer Response: If the source of the packet is NCPROUTE, contact the IBM Software Support Center. Otherwise locate the router and take corrective action.

Source Data Set: NRTRACE

Procedure Name: dumppacket Routing Information Protocol (RIP)

EZB4048E (truncated record, len *len*)

Explanation: A Routing Information Protocol (RIP) datagram was received that did not end on a route boundary. Either the packet was too large, or it has been truncated.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the packet received was built correctly and can be processed with other routers. Contact the IBM Software Support Center if the problem appears to be with NCPROUTE.

Source Data Set: NRTRACE

Procedure Name: dumppacket

EZB4049I destination *destination* metric *metric*

Explanation: A route to the destination at the indicated metric is being displayed from NCPROUTE or an adjacent router depending on the contents of the last EZB4045I message. The metrics displayed to NCPROUTE by other routers do not have the interface metrics added to them, this is done when the route is added to NCPROUTE's tables. Likewise, NCPROUTE displays routes at a metric which does not include the metric of the interface on which the route is being displayed. A route is never displayed on the interface from which it was received.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumppacket

EZB4050I (request for full tables)

Explanation: A request for a complete routing table is being sent, or has been received depending on the contents of the last EZB4045I message. This message is sent during session initialization by NCPROUTE. In addition to foreign routers making this request, application programs can ask a client for its tables by sending a Routing Information Protocol (RIP) request for a route with an address family of 0 and a metric of 16 to a client NCP at the current *RouteD* port (by default, 520).

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumppacket

EZB4051E unknown address family *family* metric *metric*

Explanation: A route in an unacceptable address family has been received, or is about to be sent depending on the contents of the last EZB4045I message. Currently, only AF_INET is supported, although AF_UNSPEC is allowed when making requests for full tables.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If the route is being sent from NCPROUTE, contact the IBM Software Support Center. Otherwise locate the router listed in the last EZB4045I message and correct it so that noninternet routes are not sent to the client NCP.

Source Data Set: NRTRACE

Procedure Name: dumppacket

EZB4052I traceon file = *dataset*

Explanation: A TRACEON packet was received. Tracing is requested to go to the specified data set.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: dumppacket

EZB4055I attempting to (re)start SNMP connection

Explanation: NCPROUTE is attempting to establish a connection with the SNMP agent specified in the profile. This occurs during startup, and at any time when an SNMP request occurs and no connection exists with the agent.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4056W no response from agent on *host*

Explanation: NCPROUTE was unable to establish a connection with the SNMP agent on host *host*.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that an agent is running on *host*, and that IP routes exist between the local host and *host*. When tracing is activated on the SNMP agent, DPI requests should be seen coming from NCPROUTE.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4057W Start the agent before issuing SNMP queries.

Explanation: A reminder that the SNMP agent should be running before NCPROUTE will be able to process SNMP queries for its clients.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4058E An error occurred while opening the SNMP socket:

Explanation: An error occurred while attempting to open a socket for communicating with the SNMP agent. A more detailed tcperror() library message follows.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the problem as indicated by the error in the detailed tcperror() library message. Refer to the *OS/390 C/C++ Run-Time Library Reference* for more information about socket() function errors.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4059I Connecting to agent *agent* **on DPI port** *port*

Explanation: NCPROUTE is attempting to connect to the SNMP agent using the DPI port.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4060E An error occurred while connecting to the SNMP agent:

Explanation: NCPROUTE was unsuccessful in connecting to the SNMP agent. A more specific error message will follow.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4061E A connection is required for processing SNMP requests.

Explanation: A reminder that the SNMP agent socket must be connected before SNMP queries are processed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4062I SNMP DPI connection established

Explanation: The connection with the SNMP agent is established and SNMP queries will now be processed as they arrive.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4063E Unable to register *root* **with SNMP agent.**

Explanation: NCPROUTE was unable to register the MIB extension root with the agent. An I/O error occurred while sending the registration to the agent. The connection to the agent is closed and will be reopened for the next query.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the agent is operating correctly and that the agent's host is currently reachable from the local host.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4064I *root* **registered with SNMP agent**

Explanation: The MIB extension root has been registered with the SNMP agent.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: init_snmp

EZB4065E Unable to establish a session with the SNMP agent.

Explanation: An attempt to reestablish a connection with the SNMP agent has failed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the SNMP agent is running. Verify that the SNMP statements in the NCPROUTE.PROFILE are correct, and verify that the host running the SNMP agent can be reached from the local host.

Source Data Set: NRSNMP

Procedure Name: snmp_input

EZB4066E An incoming SNMP packet was discarded (noagent)

Explanation: Because no connection exists with the SNMP agent, the SNMP packet being processed is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_input

EZB4067E Error while forwarding SNMP packet to agent:

Explanation: NCPROUTE was unable to forward the SNMP request up to the SNMP agent. A more specific error message follows. The SNMP packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the SNMP agent is running and that the host running the agent is reachable from the local host.

Source Data Set: NRSNMP

Procedure Name: snmp_input

EZB4068I SNMP response from local agent

Explanation: The SNMP agent has responded to a forwarded request.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_response

EZB4069E Error while sending SNMP reply to client *client*

Explanation: NCPROUTE was unable to send the response to a SNMP query back to the client. The packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the following error message, and take whatever corrective action is recommended.

Source Data Set: NRSNMP

Procedure Name: snmp_response

EZB4070S Couldn't parse incoming DPI packet! (dropping) This suggests a problem with the SNMP agent.

Explanation: NCPROUTE could not parse an incoming DPI packet from the SNMP agent. The packet is damaged or has not been built properly. Either the SNMP agent or the SNMP dpi library is in error. Also, they may be at different maintenance levels.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4071I *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 NCP client's gateways data set for an interface or in the NCPROUTE profile data set. If there are no RIP control settings for an interface, NCPROUTE will use the one from the profile settings.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT, NROUTPUT

Procedure Name: rip_input, supply

EZB4072I SNMP: DPI GET request (*oid*) received.

Explanation: An SNMP DPI get request for the variable specified by the ASN.1 object identifier was received. The object identifier is a registered MIB extension, and if you remove the root of the object identifier (based on the contents of EZB4064I) you will have a client identifier followed by the original object identifier.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4073I SNMP sub-agent: DPI GET NEXT request (*oid*) received.

Explanation: An SNMP DPI get next request for the specified object identifier has been received.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4074I SNMP sub-agent: DPI SET request received

Explanation: An SNMP DPI set request was received. SNMP set commands are not supported in this release.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4075S Unexpected SNMP query type *type*; SNMP support appears incomplete in NCPROUTE.

Explanation: An unexpected SNMP DPI packet type has been received from the SNMP agent. The packet has already been validated by the DPI library routines, so this indicates an unsupported valid DPI packet type. An error is returned to the originator of the SNMP packet.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4076I RIP2 packet from router *router* not authorized

Explanation: A RIP Version 2 packet, received from *router*, is ignored due to a authentication key mismatch. Authentication is enabled for RIP Version 2 packets according to the interface options in the NCP client's gateways data set or in the NCPROUTE profile data set. If there are no interface settings, NCPROUTE will use the one from the profile settings.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB4077S SNMP sub-agent: received DPI request outside tree

Explanation: The SNMP agent has requested information or action on a MIB variable that is outside of the tree that NCPROUTE manages. This indicates an error in the agent code, because NCPROUTE did not register this variable. NO_SUCH_NAME is returned.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Compare the object identifier for the current DPI packet with the region of the MIB tree that NCPROUTE manages. The object identifier can be found in the last EZB4072I, EZB4073I, or EZB4074I message issued. The registered region can be found in the last EZB4064I message issued.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4078I SNMP sub-agent: received invalid DPI request outside tree

Explanation: The DPI packet from the agent refers to an OID which is not present in the managed tree.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4079I iproutedest.*instance*

Explanation: A request has been made for the destination of the specified route. Each route in the client's table is numbered starting with 0.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4080I iprouteifindex.destination

Explanation: A request has been made for the interface index associated with the route to *destination* for the client which forwarded this request to NCPROUTE.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4081I iproutemetric1. *destination*

Explanation: A request has been made for the metric associated with the route to the destination for the client that forwarded this request to NCPROUTE.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4082I iproutemetric(2-4).*destination* (unsupported)

Explanation: A request was made for one of the alternate route metrics that is not used under RIP. If a route to the specified destination exists, a value of '-1' is returned; otherwise, NO_SUCH_NAME will be returned.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4083I *iproutenexthop.instance*

Explanation: A request was made for the next hop for the specified route. Routes are specified by the instance number, which is the ordinal position within NCPROUTE's tables. If the specified instance exists, an internet address will be returned; otherwise, NO_SUCH_NAME will be returned.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4084I *iproutetype.instance*

Explanation: A request was made for the route type for the specified route. If the specified instance exists, one of three values will be returned depending on the route state:

Value	Explanation
Direct	The route is to a directly connected destination.
Invalid	The route has an infinite metric.
Remote	The route is to an indirectly connected destination requiring one or more routers to reach.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4085I *iprouteproto.instance*

Explanation: A request was made for the mechanism by which the route for the specified instance was determined. Values returned will be 'RIP' or 'Other' depending on whether the route was learned dynamically or was manually entered.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4086I *iprouteage.instance*

Explanation: A request was made for the age of the route of the specified instance. The value returned will be the length of time the route has been active.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4087I *iproutemask.instance*

Explanation: A request was made for the network or subnetwork mask of route of the specified instance. The value returned will be the network or subnetwork mask value in dotted-decimal notation.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4088E A request was made for an unsupported MIB variable: *variable*

Explanation: A request was made for an unsupported MIB variable. Only variables in the iproute group are returned. NO_SUCH_NAME will be returned for the unsupported MIB variable.

System Action: NCPROUTE continues.

User or Operator Response: Reissue the request with a valid MIB variable from the iproute group.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4089E An attempt was made to change a MIB variable. SNMP set request is not currently supported.

Explanation: An attempt was made to change a MIB variable. Variables in the iproute group may be queried but not changed. An SNMP_GEN_ERR is returned.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: dpi_set

EZB4090E SNMP buffer overrun: *number* **bytes**

Explanation: An SNMP request will exceed NCPROUTE's internal buffer size during an edit. This almost always indicates an incorrect request because no supported object identifier is large enough to cause this. The buffer is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify the object identifier being requested is for an iproute MIB variable. If it is, contact the IBM Software Support Center.

Source Data Set: NRSNMP

Procedure Name: edit_obj

EZB4125I ASN.1 type at *count* bytes into packet: Class=*class* Field=*field*

Explanation: An ASN.1 object such as an object identifier or an integer was located in the SNMP packet being displayed. The count is the offset into the packet where the object was located. The object's tag's class bits are formatted and displayed as *class.* An object's class can be viewed as the scope of the object identifier. ASN.1 defines four values:

Value Explanation

Universal
Well-known tags. NCPROUTE wants this type of tags.

Application-Wide
Tags local to the application. NCPROUTE does not have any of these.

Context-Specific
Used in ASN.1 constructors. Not applicable to NCPROUTE.

Private-Use
NCPROUTE does not use any private tags.

The object's tag's field bits are formatted and displayed as the field. The field is either primitive or constructed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: decode

EZB4126E Modify command ignored, invalid client address

Explanation: The target client specified in the MODIFY command is not valid.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the address for the client target.

Source Data Set: NRMAIN

Procedure Name: do_modify

EZB4127E Modify command ignored, no session with client *client*

Explanation: The target client specified in the MODIFY command is unknown. No session exists between NCPROUTE and the client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Wait for NCPROUTE to establish a session with the NCP client or respecify the target client using an address known by NCPROUTE.

Source Data Set: NRMAIN

Procedure Name: do_modify

EZB4128I Reserved for addenda (*type*)

Explanation: The type value is not a known object tag type. It was described as reserved for addenda in RFC 1158.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: decode

EZB4129I *type*

Explanation: The formatted ASN.1 object tag's number. NCPROUTE formats Universal tags as defined in RFC 1158.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: decode

EZB4130I Misc type *type*

Explanation: This object's class indicates that it is not a well-known type. The unformatted object type is displayed, NCPROUTE only formats objects with a class of Universal.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: decode

EZB4131I Encountered editable OID at offset *offset*

Explanation: An object identifier was located at the indicated offset into the packet. NCPROUTE will need to translate this OID to another region of the MIB tree so that the agent will recognize the request as being for one of NCPROUTE's clients and not for the host running the agent.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: decode

EZB4132I Length = *length*

Explanation: The length of the decoded ASN.1 object is displayed.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: decode

EZB4133I Adding *type route* **route** *destination* **via gateway** *gateway*, **metric** *metric*

Explanation: The indicated route, defined in the client's *hlq.*.ETC.GATEWAYS data set, is added to the NCP's IP routing table. The route to the gateway will not be replaced by a competing RIP route.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4134W Subnetwork mask unknown for *destination*, **using network route** *route*

Explanation: The indicated subnetwork route, defined in the client's GATEWAYS data set, was explicitly coded as a "net" route type. Because the subnetwork mask for the destination subnetwork is unknown, NCPROUTE replaces the subnetwork route with a network route. NCPROUTE currently does not support variable subnetting.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTABLES

Procedure Name: nradd

EZB4135E Invalid host or (sub)network address '*destination*'

Explanation: In the line entry for the client's *hlq.*.ETC.GATEWAYS data set, the gateway definition has an incorrect destination address. The *destination* must be either a resolvable host name or an internet address in dotted notation. The gateway entry is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*.ETC.GATEWAYS data set.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4136W Second element ignored, changing to 'active'

Explanation: The second element in the active gateway entry is detected to be incorrect. NCPROUTE will change the element to be ACTIVE since it is likely that the gateway entry is in error. With this change, the active gateway entry is processed normally.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*.ETC.GATEWAYS data set.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4138W Unknown next hop address *ipaddr* **for route** *destination* **from router** *router*

Explanation: An unknown next hop address *ipaddr* was received in a RIP packet for a route to *destination* from from *router*. The route is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Locate the router that produced the packet and correct the problem. It may involve router reconfiguration.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB4139S tblclr: no host hash table for client *client*

Explanation: A required host hash table could not be located for the client. NCPROUTE was attempting to remove all routes for the specified client from its tables. Any routes will be left in NCPROUTE's tables.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: tblclr

EZB4141S tblclr: no network hash table for client *client*

Explanation: A required network hash table could not be located for the client, but the host hash table has already been found and cleared. NCPROUTE was attempting to remove all network routes for the client from its tables. Any network routes for the client remain in NCPROUTE's tables.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRTABLES

Procedure Name: tblclr

EZB4142E The transmission of an 'Ack' to client *client* **failed.**

Explanation: TCPIP detected that an "Acknowledge" PDU could not be delivered successfully before any transmission was performed. The PDU is discarded. A more specific error message will immediately follow.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Examine the following error message and follow the documented procedures.

Source Data Set: NRPDUS

Procedure Name: send_ack

EZB4143S No SNMP_AGENT statement in profile

Explanation: A required SNMP_AGENT statement was missing from the NCPROUTE profile. This statement must identify the host which runs the SNMP agent which NCPROUTE will use to resolve queries. SNMP requests will not be honored by NCPROUTE (or its clients).

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the profile.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB4144S SNMP requests will not be honored.

Explanation: One or more severe errors were encountered which will result in a connection with the SNMP agent not being established. Until these errors are resolved, and NCPROUTE is restarted, SNMP requests will not be honored by NCPROUTE and its clients in turn.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct any previously identified errors.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB4145S * No SNMP_COMMUNITY statement in NCPROUTE profile

Explanation: A required SNMP_COMMUNITY statement is missing from the NCPROUTE profile. This statement identifies the SNMP community which will be used when forwarding SNMP requests to the agent. SNMP requests will not be honored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the profile and restart NCPROUTE.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB4146W Required NCPROUTE profile dataset (*profile*) not found

Explanation: The optional profile configuration data set for NCPROUTE could not be opened successfully. This message indicates which data set the open was attempted on.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify that the profile data set is defined in //DD:NCPRPROF of the NCPROUTE start proc JCL and is accessible by NCPROUTE. If the data set is sequential, ensure that the FREE=CLOSE parameter is specified. Do not specify this parameter if the data set is partitioned.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB4147S Unknown keyword (*keyword*) in PROFILE

Explanation: An unknown keyword *keyword* was used in a statement in the NCPROUTE profile. The statement is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the NCPROUTE profile.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB4148E client *client*: failed attempt to add route to *destination*.

Explanation: NCPROUTE's attempt to add a route to the client was rejected by the client. Most likely the client's route tables are full so that it could not accept any more routes, or the NCP is attempting to add a route specified as PERM during generation.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify the routing table size values in the NUMROUTE keyword in the IPOWNER statement of the NCP generation. Increase the values, if necessary.

Source Data Set: NRPDUS

Procedure Name: add_fail

EZB4149E client *client*: failed attempt to delete route to *destination*.

Explanation: Client *client* rejected NCPROUTE's attempt to delete a route to *destination* from the client's tables. Most likely the route did not exist in the client's tables.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: delete_fail

EZB4150I End of GATEWAYS processing

Explanation: Processing is completed for the GATEWAYS data set member.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4151T Out of memory while processing GATEWAYS

Explanation: NCPROUTE has exhausted available storage while processing a client's GATEWAYS data set member.

System Action: NCPROUTE ends abnormally.

User or Operator Response: None.

System Programmer Response: Increase NCPROUTE's region size and restart. Take into consideration that storage requirements are based on the number of clients being served and the number of routes being managed. If the problem still cannot be resolved, contact the IBM Software Support Center.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4152I Adding active gateway *ip_addr*, metric *metric*

Explanation: The indicated active gateway, which is defined in the client's *hlq*..ETC.GATEWAYS data set, is added to NCPROUTE's routing table. The route to the active gateway will be treated as a network interface.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4154E Invalid option: *option*

Explanation: In the line entry for the client's *hlq.*.ETC.GATEWAYS data set, the NCPROUTE's server options definition has an incorrect option. Although other options may be processed normally, the incorrect option is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*.ETC.GATEWAYS data set. See the *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4155E Invalid default router value: *value*

Explanation: In the line entry for the client's *hlq.*.ETC.GATEWAYS data set, the NCPROUTE's server options definition has an incorrect default router value. The incorrect default router value is ignored and the default value is used.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*.ETC.GATEWAYS data set by specifying a valid default router value. See the *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4156E Invalid trace level: *level*

Explanation: In the line entry for the client's *hlq.*.ETC.GATEWAYS data set, the NCPROUTE's server options definition has an incorrect trace level. The incorrect trace level is ignored and the default value is used.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*.ETC.GATEWAYS data set by specifying a valid trace level value. See the *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4157E Invalid *type* **value:** *value*

Explanation: In the line entry for a client's *hlq.*ETC.GATEWAYS data set, the options definition contains an invalid value. Although other options may be processed normally, the invalid option is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the client's *hlq.*ETC.GATEWAYS data set. See *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4158I Global tracing at all levels suppressed

Explanation: Trace levels for all NCP clients are not displayed. This includes the trace levels specified in the clients' *hlq..*ETC.GATEWAYS data set.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: global_trace_start

EZB4159I Global tracing actions started

Explanation: Trace levels for all NCP clients have been advanced to the ″actions″ level, which causes messages to 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. The trace levels specified in the client's *hlq..*ETC.GATEWAYS data sets are also advanced to this level.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: global_trace_start

EZB4160I Global tracing packets started

Explanation: Trace levels for all NCP clients have been advanced to the ″packets″ level, which displays the types of packets sent and received in addition to the output displayed at the ″actions″ level. The trace levels specified in the client's *hlq..*ETC.GATEWAYS data sets are also advanced to this level.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRTRACE

Procedure Name: global_trace_start

EZB4161T Global trace levels exceeded maximum of 2 -t's. For each client, use the MODIFY command or GATEWAYS dataset to specify additional tracing options.

Explanation: An incorrect number of trace levels (-t's) was passed from *hlq.*SEZAINST(NCPROUTE) start proc JCL. The parameter(s) could either be passed from the command line parameters or from the default parameter list in the start proc JCL.

System Action: NCPROUTE exits.

User or Operator Response: None.

System Programmer Response: Specify a correct number of *-t*'s in the command line parameters or in the default parameter list of the start proc JCL. If a higher trace level is desired, specify the trace level in the options statement of a client's *hlq..*ETC.GATEWAYS data set. Another option is to use the MODIFY command to increase the trace levels for the client after NCPROUTE has started and established a session with the client. Refer to *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

Source Data Set: NRMAIN

Procedure Name: parse_parms

EZB4162I Deferring add route to *destination*

Explanation: The addition of a new route to the specified destination in the NCP client's routing table is deferred until after the NCP client has finished initialization. The new route is copied to a static buffer, which is used after the Inactive Interface List PDU has been received from the NCP client. NCPROUTE will perform the add from the static buffer.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: defer_add

EZB4163E Too many deferred routes

Explanation: The number of static buffers to hold the new routes for deferred addition to the NCP clients' routing tables has exceeded the maximum of 500. This may happen when multiple NCP clients have not completed their initialization processes at the same time. At this time, the new routes are not added to the NCP client's routing tables. The static buffers do not become available until after NCPROUTE has received the Inactive Interface List PDUs from the NCP clients. These PDUs tells NCPROUTE that the NCP clients have completed their initialization processes.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Allow sufficient time for the NCP clients to complete their initialization processes so that the new routes can be added to the client's routing tables. Contact the IBM Software Support Center if the problem persists.

Source Data Set: NRPDUS

Procedure Name: defer_add

EZB4164I Init: compare with *client*

Explanation: The client's internet address is being compared with one of NCPROUTE's session table entries for an exact match.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: ncp_initialized

EZB4165E Client *client* **not in session table**

Explanation: The client's internet address was not found in the session table. This indicates that NCPROUTE has not established a session with the NCP client. NCPROUTE will not manage the routes for the client until a ″Hello″ PDU has been received for session establishment.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Allow sufficient time for the NCP client to issue the ″Hello″ PDUs so that a session can be established with NCPROUTE. If a Hello PDU was received and the problem persists, contact the IBM Software Support Center.

Source Data Set: NRPDUS

Procedure Name: ncp_initialized

EZB4166I Session with client *client* started

Explanation: NCPROUTE has successfully established a session with the specified NCP client through the handshaking process and is waiting to receive an Inactive Interface List PDU from the client. This PDU indicates that the client has completed initialization and is ready for route table management by NCPROUTE.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRPDUS

Procedure Name: recv_hello

EZB4167E SNMP requester *requester* is not reachable by NCP client *client*

Explanation: NCPROUTE was ready to transport the SNMP response packet to the NCP client to be forwarded to the SNMP requester, but the route to the SNMP requester was detected to be unavailable. A possible cause is that the NCP client's interface to the SNMP requester became inactive. The SNMP response packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify the status of the NCP client's interface to the SNMP requester. If the interface was inactive, activate the interface so that a route can be created; otherwise, determine whether the route to the SNMP requester was timed out. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRSNMP

Procedure Name: snmp_response

EZB4168I SNMP lookup failed: *reason*

Explanation: NCPROUTE could not obtain the routing information based upon the SNMP request packet sent by the SNMP requester. In this case, NO_SUCH_NAME is returned to the SNMP requester. Reasons for the failure are:

Reason Explanation

NO INSTANCES PROVIDED
The target route was not specified in the SNMP request; however, this is an exception to SNMP GET_NEXT requests.

ROUTE TABLE EMPTY
There were no routing table entries for this interface based upon the target route.

ROUTE NOT FOUND
The target route was not found in the routing table.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Verify the accuracy of the target route in the SNMP request. If the problem cannot be corrected, contact the IBM Software Support Center.

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4172I SNMP reply sent to NCP client *client*

Explanation: NCPROUTE is sending the reply containing the SNMP response packet to the NCP client to be forwarded to the SNMP requester.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_response <

EZB4173I SNMP get_next: no session with client *client*

Explanation: During processing of the SNMP GET_NEXT request, NCPROUTE has detected that there was no session with the NCP client. The SNMP request packet is discarded.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: If the session with the NCP client appears to have terminated, examine previous messages to determine the cause. When necessary, take corrective actions to allow NCPROUTE to process the SNMP requests for the NCP client.

Source Data Set: NRSNMP

Procedure Name: get_next

EZB4174I SNMP get_next: searching for next route after *target*

Explanation: During processing of the SNMP GET_NEXT request, NCPROUTE is searching for the next route after the target route in the routing tables for the NCP client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: get_next

EZB4175I SNMP get_next: found next route *ip_addr*

Explanation: During processing of the SNMP GET_NEXT request, NCPROUTE has found the next route entry after the target route in the routing tables for the NCP client. The entry contains the next route's *ip_addr*.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: get_next

EZB4176I SNMP get_next: compare with *target*

Explanation: The target route specified in the SNMP GET_NEXT request packet received by NCPROUTE from an NCP client is being compared with one of NCPROUTE's routing table entries for an exact match.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: get_next

EZB4177I SNMP get_next: addresses matched

Explanation: A match was found for the target route specified in the SNMP GET_NEXT request packet with one of NCPROUTE's routing table entries for an NCP client.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: get_next

EZB4178I RIP2 authentication *action* **at** *level* **level (***interface***)**

Explanation: RIP Version 2 authentication is enabled or disabled at the specified level for an interface for for all interfaces in the NCP client, or for all NCP clients.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN, NRSTART

Procedure Name: read_profile, ParseOptions

EZB4179E SNMP object length exceeded maximum 64K

Explanation: During the object data conversion process for an SNMP packet, NCPROUTE detected an incorrect length value in the header portion of the object data. NCPROUTE could not continue processing for the value has exceeded the maximum length of 64K. The SNMP packet may have been built incorrectly.

System Action: NCPROUTE exits.

User or Operator Response: None.

System Programmer Response: Contact your IBM Software Support Center.

Source Data Set: NRSNMP

Procedure Name: store_int

EZB4180I Packet from router *router* **ignored (filtered out)**

Explanation: NCPROUTE is ignoring the RIP packet as a result of being filtered out according to a RIP input or output filter defined in the NCP clients's gateways data set.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRINPUT

Procedure Name: rip_input

EZB4181I Interface *interface* **skipped, interface is multicast-incapable**

Explanation: The interface *interface* is skipped because the interface is not capable of multicasting RIP Version 2 packets. The RIP2 supply control has been configured for the interface and disallows the use of broadcasting RIP packets over the interface.

System Action: NCPROUTE 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: NROUTPUT

Procedure Name: toall_ifs

EZB4182I SNMP request received from NCP client *client*

Explanation: An SNMP query request was received from an NCP client for processing. The NCP client had received the query request from its SNMP client or the SNMP requester.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: snmp_input

EZB4191I SNMP get-next: next route not found (*reason***)**

Explanation: NCPROUTE is processing the SNMP GET_NEXT request but it could not obtain the routing information for the next route. In this case, NO_SUCH_NAME is returned to the SNMP requester. Reasons for the unavailable route are:

Reason Explanation

NOT IN TABLE
The target route was not found in the routing table. This implies that the route entry for the next route cannot be determined.

END OF TABLE
The target route was found at the end of the routing table but there were no more route entries for the next route.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: These reasons require the following responses:

Reason Response

NOT IN TABLE
Verify the accuracy of the target route in the SNMP request.

END OF TABLE
None.

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

Source Data Set: NRSNMP

Procedure Name: snmp_lookup

EZB4194I SNMP sub-agent received DPI request

Explanation: NCPROUTE is performing the role as an SNMP sub-agent and is in session with the SNMP agent over the Distributed Program Interface (DPI). NCPROUTE has received a DPI request from the SNMP agent for processing.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSNMP

Procedure Name: dpi_in

EZB4195I Option(s): *options*

Explanation: Additional NCPROUTE options, specified in a client's *hlq.*.ETC.GATEWAYS data set member, are being processed. Some options may be overridden by the parameter list in the *hlq.*.SEZAINST(NCPROUT) start proc JCL. See the *OS/390 IBM Communications Server: IP Configuration Reference* for more information.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4196I Opening NCPROUTE profile data set *datasetname*

Explanation: The specified NCPROUTE profile data set is being opened. Entries in the data set are read in for input.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRMAIN

Procedure Name: read_profile

EZB4197E Invalid route type '*type***'**

Explanation: In the line entry for the NCP client's *hlq.*.ETC.GATEWAYS data set, the gateway definition has an incorrect route type. Allowable route types are HOST for host route, NET for network or subnetwork route, and ACITVE for a route to be treated as a network interface. The gateway entry is ignored.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: Correct the NCP client's *hlq.*.ETC.GATEWAYS data set.

Source Data Set: NRSTART

Procedure Name: gateways

EZB4198I (no etc.gateway definitions)

Explanation: The NCP client's *hlq.*.ETC.GATEWAYS data set contains no optional gateway definitions.

System Action: NCPROUTE continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: NRSTART

Procedure Name: gateways

Chapter 8. EZB6473I

Initialization Message

This section contains an initialization message.

EZB6473I TCP/IP STACK FUNCTIONS INITIALIZATION COMPLETE.

Explanation: TCP/IP has been successfully initialized.

System Action: TCP/IP continues.

User or Operator Response: None.

System Programmer Response: None.

Source Data Set: EZBTIINI

Chapter 9. EZB8801I

TCP/IP FFST Communication Error Message

This section contains a TCP/IP FFST communication error message.

EZB8801I *probenm* **ATTEMPTED, FFST NOT AVAILABLE.**

Explanation: TCP/IP encountered an anomaly and attempted to execute a FFST probe. FFST was not available to service the request. The request was handled by furnishing the identification of the probe.

In the message text:

probenm
The probe identifier.

Source Data Set: EZBITZPF

System Action: TCP/IP continues.

User or Operator Response: Determine why FFST is not available. Gather what probe documentation is available and contact your IBM support center.

System Programmer Response: Determine why FFST is not available. Gather what probe documentation is available and contact your IBM support center.

Chapter 10. EZB8500—EZB9744

Common Messages for the Variable SayCalRe

This section contains messages that are called by several application and function components. When these messages are called, only the message text will be appended to the message that called it. The message will inherit the prefix (EZA or EZB) of the message that called it. They are documented fully in *OS/390 IBM Communications Server: IP Messages Volume 1 (EZA)*.

The following is a complete list of these common messages.

- EZB9395I
- EZB9396I
- EZB9397I
- EZB9398I
- EZB9399I
- EZB9400I
- EZB9401I
- EZB9402I
- EZB9403I
- EZB9404I
- EZB9405I
- EZB9406I
- EZB9407I
- EZB9408I
- EZB9409I
- EZB9410I
- EZB9411I
- EZB9412I
- EZB9413I
- EZB9414I
- EZB9415I
- EZB9416I
- EZB9417I
- EZB9418I
- EZB9418I
- EZB9419I
- EZB9420I
- EZB9421I
- EZB9422I
- EZB9423I
- EZB9424I
- EZB9428I

Appendix A. Information Apars

This appendix lists information apars for IP-related books.

Notes:

1. Information apars contain updates to previous editions of the manuals listed below. Books updated for V2R10 contain all the updates except those contained in the information apars that may be issued after V2R10 books went to press.
2. Information apars are predefined for CS for OS/390 V2R10 and may not contain updates.

IP Information Apars

Table 2 lists information apars for IP-related books.

Table 2. IP Information Apars

Title	CS for OS/390 2.10	CS for OS/390 2.8	CS for OS/390 2.7	CS for OS/390 2.6	CS for OS/390 2.5	TCP/IP 3.3
High Speed Access Service User's Guide (GC31-8676)		ii11629	ii11566	ii11412	ii11181	
IP API Guide (SC31-8516)	II12371	ii11635	ii11558	ii11405	ii11144	
IP CICS Sockets Guide (SC31-8518)		ii11626	ii11559	ii11406	ii11145	
IP Configuration (SC31-8513)		ii11620 ii12068	ii11555 ii11637 ii11995	ii11402 ii11619 ii12066	ii11159 ii11979	ii10633
IP Configuration Guide (SC31-8725)	II12362					
IP Configuration Reference (SC31-8726)	II12363					
IP Diagnosis (SC31-8521)	II12366	ii11628	ii11565	ii11411	ii11160 ii11414	ii10637
IP Messages Volume 1 (SC31-8517)	II12367	ii11630	ii11562	ii11408		Messages and Codes ii10635
IP Messages Volume 2 (SC31-8570)	II12368	ii11631	ii11563	ii11409		
IP Messages Volume 3 (SC31-8674)	II12369	ii11632	ii11564	ii11410	ii11158	
IP Migration (SC31-8512)	II12361	ii11618	ii11554	ii11401		

Table 2. IP Information Apars (continued)

Title	CS for OS/390 2.10	CS for OS/390 2.8	CS for OS/390 2.7	CS for OS/390 2.6	CS for OS/390 2.5	TCP/IP 3.3
IP Network Print Facility (SC31-8522)		ii11627	ii11561	ii11407	ii11150	
IP Programmer's Reference (SC31-8515)		ii11634	ii11557	ii11404		ii10636
IP and SNA Codes (SC31-8571)	II12370	ii11917	Added TCP/IP codes to VTAM codes V2R6 ii11611	ii11361	ii11146 ii11097	
IP User's Guide (GC31-8514)	II12365	ii11625	ii11556	ii11403	ii11143	ii10634

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O.Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs

conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

Some updates of this book, such as an update between OS/390 releases, might be available only in softcopy. You can obtain softcopy from the OS/390 Online Library Collection (SK2T-6700), the OS/390 PDF Library Collection (SK2T-6718), or the OS/390 Internet Library (http://www.ibm.com/s390/os390/). To order the latest hardcopy edition that is available, you might need to order a lower suffix (dash) level.

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
APPN
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
Domino
DRDA
eNetwork
Enterprise Systems Architecture/370
ESA/390
ESCON
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
Parallel Sysplex
Personal System/2
PR/SM
PROFS
PS/2
RACF
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
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
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.

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/tradmarx.htm.

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

IBM Communications Server for OS/390 Publications

This bibliography contains descriptions of the books in the IBM Communications Server for OS/390 library.

Updates to books are available on RETAIN. See "Appendix A. Information Apars" on page 323 for a list of the books and the INFOAPARS associated with them.

These books are available online. Go to http://www.ibm.com/s390/os390/bkserv/ to access the OS/390 Internet Library web page.

Some books are available in both hard- and soft-copy, or soft-copy only. The following abbreviations follow each order number:

HC/SC	Both hard- and soft-copy are available
SC	Only soft-copy is available

Related Publications

For information about OS/390 products, refer to *OS/390 Information Roadmap* (GC28-1727-07 [HC/SC]). The Roadmap describes what level of documents are supplied with each release of CS for OS/390, as well as describing each OS/390 publication.

Firewall

OS/390 SecureWay Security Server Firewall Technologies Guide and Reference (SC24-5835 [HC/SC])

OSA-Express

OSA-Express Customer's Guide and Reference (SA22-7403 [HC/SC])

Softcopy Information

- *OS/390 Online Library Collection* (SK2T-6700).

  This collection contains softcopy unlicensed books for OS/390, Parallel Sysplex products, and S/390 application programs that run on OS/390. This collection is updated quarterly with any new or updated books that are available for the product libraries included in it.
- *OS/390 PDF Library Collection* (SK2T-6718).

  This collection contains the unlicensed books for OS/390 Version 2 Release 6 in Portable Document Format (PDF).
- *OS/390 Licensed Product Library* (LK2T-2499).

  This library contains unencrypted softcopy licensed books for OS/390 Version 2. If any of the books in this library are changed, it is updated quarterly. The OS/390 Licensed Product Library for Version 1 (LK2T-6702) is still available, but is no longer updated.
- *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.

Planning

OS/390 IBM Communications Server: SNA Migration (SC31-8622 [HC/SC]). This book is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.

OS/390 IBM Communications Server: IP Migration (SC31-8512 [HC/SC]). This book is intended to help you plan for IP, whether you are migrating from a previous version or installing IP for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with IP.

Resource Definition, Configuration, and Tuning

OS/390 IBM Communications Server: IP Configuration Guide (SC31-7134 [HC/SC]). This book describes the major concepts involved in understanding and configuring an IP network. Familiarity with MVS operating, IP protocols, OS/390 UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this book in conjunction with the *OS/390 IBM Communications Server: IP Configuration Reference*.

OS/390 IBM Communications Server: IP Configuration Reference (SC31-8725 [HC/SC]). This book presents information for people who

want to administer and maintain IP. Use this book in conjunction with the *OS/390 IBM Communications Server: IP Configuration Guide*. The information in this book includes:

- TCP/IP configuration data sets
- Configuration statements
- Operator commands
- Translation tables
- SMF records
- Protocol number and port assignments

OS/390 IBM Communications Server: SNA Network Implementation Guide (SC31-8563 [HC/SC]). This book presents the major concepts involved in implementing a SNA network. Use this book in conjunction with the *OS/390 IBM Communications Server: SNA Resource Definition Reference*.

OS/390 IBM Communications Server: SNA Resource Definition Reference (SC31-8565 [HC/SC]). This book describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA.Use this book in conjunction with the *OS/390 IBM Communications Server: SNA Network Implementation Guide*.

OS/390 IBM Communications Server: SNA Resource Definition Reference (SC31-8566 [HC/SC]). This book contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.

OS/390 eNetwork Communications Server: AnyNet SNA over TCP/IP (SC31-8578 [SC]). This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.

OS/390 eNetwork Communications Server: AnyNet Sockets over SNA (SC31-8577 [SC]). 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.

Operation

OS/390 IBM Communications Server: IP User's Guide (SC31-7136 [HC/SC]). This book is for people who want to use TCP/IP for data communication activities such as FTP and Telnet. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

OS/390 IBM Communications Server: SNA Operation (SC31-8567 [HC/SC]). This book serves as a reference for programmers and operators requiring detailed information about specific operator commands.

OS/390 IBM Communications Server: Quick Reference (SX75-0121 [HC/SC]). This book contains essential information about SNA and IP commands.

OS/390 eNetwork Communications Server: High Speed Access Services Users Guide (GC31-8676 [SC]). This book is for end users and system administrators who want to use applications using a High Speed Access Services connection available in CS for OS/390.

Customization

OS/390 IBM Communications Server: SNA Customization (LY43-0110 [SC]). This book enables you to customize SNA, and includes:

- 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

OS/390 eNetwork Communications Server: IP Network Print Facility (SC31-8074 [SC]). This book 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.

Writing Application Programs

OS/390 IBM Communications Server: IP Application Programming Interface Guide (SC31-7187 [SC]). This book describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.

OS/390 IBM Communications Server: IP CICS Sockets Guide (SC31-8518 [SC]). This book is for people who want to set up, write application

programs for, and diagnose problems with the socket interface for CICS using TCP/IP for MVS.

OS/390 eNetwork Communications Server: IP IMS Sockets Guide (SC31-8519 [SC]). This book is for programmers who want application programs that use the IMS TCP/IP application development services provided by IBM TCP/IP for MVS.

OS/390 IBM Communications Server: IP Programmer's Reference (SC31-8515 [SC]). This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

OS/390 IBM Communications Server: SNA Programming (SC31-8573 [SC]). This book 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.

OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Guide (SC31-8581 [SC]). This book describes how to use the SNA LU 6.2 application programming interface for host application programs. This book 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 book.)

OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference (SC31-8568 [SC]). This book provides reference material for the SNA LU 6.2 programming interface for host application programs.

OS/390 eNetwork Communications Server: CSM Guide (SC31-8575 [SC]). This book describes how applications use the communications storage manager.

OS/390 IBM Communications Server: CMIP Services and Topology Agent Guide (SC31-8576 [SC]). This book describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The book provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis

OS/390 IBM Communications Server: IP Diagnosis (LY43-0105 [HC/SC]). This book 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.

OS/390 IBM Communications Server: SNA Diagnosis V1 Techniques and Procedures(LY43-0079 [HC/SC]) and *OS/390 IBM Communications Server: SNA Diagnosis V2 FFST Dumps and the VIT* (LY43-0080 [HC/SC]). These books help you identify a 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.

OS/390 IBM Communications Server: SNA Data Areas Volume 1 (LY43-0111 [SC]) and *OS/390 IBM Communications Server: SNA Data Areas Volume 2* (LY43-0112 [SC]). These books describe SNA data areas and can be used to read a SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and Codes

OS/390 IBM Communications Server: SNA Messages (SC31-8569 [HC/SC]). This book describes the ELM, IKT, IST, ISU, IVT, IUT, and USS messages. Other information in this book includes:

- Command and RU types in SNA messages
- Node and ID types in SNA messages
- Supplemental message-related information

OS/390 IBM Communications Server: IP Messages Volume 1 (EZA) (SC31-8517 [HC/SC]). This volume contains TCP/IP messages beginning with EZA.

OS/390 IBM Communications Server: IP Messages Volume 2 (EZB) (SC31-8570 [HC/SC]). This volume contains TCP/IP messages beginning with EZB.

OS/390 IBM Communications Server: IP Messages Volume 3 (EZY-EZZ-SNM) (SC31-8674 [HC/SC]). This volume contains TCP/IP messages beginning with EZY, EZZ, and SNM.

OS/390 IBM Communications Server: IP and SNA Codes (SC31-8571 [HC/SC]). This book describes codes and other information that display in CS for OS/390 messages.

APPC Application Suite

OS/390 eNetwork Communications Server: APPC Application Suite User's Guide (GC31-8619 [SC]). This book 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.

OS/390 eNetwork Communications Server: APPC Application Suite Administration (SC31-8620 [SC]). This book contains the information that administrators need to configure the APPC application suite and to manage the APING, ANAME, AFTP, and A3270 servers.

OS/390 eNetwork Communications Server: APPC Application Suite Programming (SC31-8621 [SC]). This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Multiprotocol Transport Networking (MPTN) Architecture Publications

Following are selected publications for MPTN:

Networking Blueprint Executive Overview (GC31-7057)

Multiprotocol Transport Networking: Technical Overview (GC31-7073)

Multiprotocol Transport Networking: Formats (GC31-7074)

Redbooks

The following Redbooks may help you as you implement CS for OS/390.

- *OS/390 eNetwork Communication Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing* (SG24–5227–01).

  This book provides examples of how to configure the base TCP/IP stack, routing daemons and the TELNET server. This book also provides information about national language support (NLS), routing, OSPF, network interfaces, diagnosis, multicasting, OS/390 UNIX System Services and security in an OS/390 UNIX System Services environment.
- *OS/390 eNetwork Communication Server V2R7 TCP/IP Implementation Guide Volume 2: UNIX Applications* (SG24–5228–01).

  This book provides information about implementing applications that run in the OS/390 UNIX environment, such as FTP, SNMP, BIND-based name server, DHCP, and SENDMAIL. This book also provides configuration samples and describes the implementation process.
- *OS/390 eNetwork Communication Server V2R7 TCP/IP Implementation Guide Volume 3: MVS Applications* (SG24–5229–01).

  This book provides information about TCP/IP applications that run in a legacy MVS environment, including CICS/IMS Sockets, and printing (NPF, LPR, and LPD.)
- *OS/390 Secureway Communication Server V2R8 TCP/IP Guide to Enhancements* (SG24-5631-00).

  This redbook provides information to facilitate the configuration and use of the new technologies and functions supported in SecureWay Communications Server for OS/390 V2R8. Special areas of interest in this book are security and Quality of Services.
- *TCP/IP in a Sysplex* (SG24–5235–01).

  The main goals of a Parallel Sysplex are high availability and high performance. This book demonstrates how these goals can be achieved in the particular environment of SecureWay Communications Server for OS/390 and its TCP/IP applications. This book describes the WLM/DNS functions, the Network Dispatcher and the Dynamic VIPA.
- *SNA and TCP/IP Integration* (SG24–5291–00).

  This book provides information about integrating current SNA network with future TCP/IP and Web-based communication requirements. This book concentrates on routing techniques.
- *SNA in a Parallel Sysplex Environment* (SG24–2113–01).

This book provides information about implementing a VTAM-based network on a Parallel Sysplex.

- *Subarea to APPN Migration : VTAM and APPN Implementation* (SG24–4656–01).

  This book is the first of two volumes. This book provides information about the migration of a subarea network to an APPN network. Some knowledge of SNA subarea networks and familiarity with the functions, terms and data flows of APPN networks is assumed.

- *Subarea to APPN Migration : HPR and DLUR Implementation* (SG24–5204–00).

  This book is the second of two volumes. This book provides information about the coverage of a network using HPR, DLUR and APPN/HPR routers. Some knowledge of SNA subarea networks and familiarity with the functions, terms and data flows of APPN networks is assumed.

Readers' Comments — We'd Like to Hear from You

OS/390 IBM Communications Server
IP Messages:
Volume 2 (EZB)
Version 2 Release 10

Publication No. SC31-8570-05

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
SC31-8570-05

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

SC31-8570-05

Cut or Fold
Along Line

Program Number: 5647-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8570-05

Spine information:

IBM

OS/390 IBM Communications Server

OS/390 V2R10.0 IBM CS IP Msgs V2 (EZB)

Version 2
Release 10