z/OS Communications Server

IP System Administrator's Commands

Version 2 Release 1

SC27-3661-02

Note:
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 1101.

This edition applies to Version 2 Release 1 of z/OS (5650-ZOS), and to subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send us comments electronically by using one of the following methods:

Internet email:
comsvrcf@us.ibm.com

World Wide Web:
http://www.ibm.com/systems/z/os/zos/webqs.html

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

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

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

Contents

Chapter 5. Displaying policy-based networking information 819

Chapter 6. Querying and administrating a Domain Name System (DNS) 891

Chapter 7. Managing TCP/IP network resources with SNMP 953

##

##

##

##

##

##

##

##

##

##

##

##

##

Figures

Tables

About this document

This document describes how to monitor the network, manage resources, and maintain performance of z/OS® Communications Server. This includes the ability to perform the following functions:

- Configure a system, using TSO and MVS™ commands
- Monitor the network
- Query name servers
- Manage network resources

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

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

A companion to this document is the z/OS Communications Server: IP User's Guide and Commands, which describes how to use the applications available in z/OS Communications Server V2R1.

Who should read this document

This document is written for system administrators who need to understand how to monitor applications and network resources provided by z/OS Communications Server V2R1.

Before using this document, you should be familiar with the IBM® Multiple Virtual Storage (MVS) operating system, the IBM Time Sharing Option (TSO), and z/OS UNIX System Services and the z/OS UNIX shell. In addition, z/OS Communications Server V2R1 should already be installed and customized for your network. For information about installing, see the z/OS V2R1 Program Directory. For information about customizing, see the z/OS Communications Server: IP Configuration Reference.

How this document is organized

This document contains the following information:

- Chapter 1, “Operator commands and system administration,” on page 1 is a reference of commonly used commands for experienced system programmers.
- Chapter 2, “Sending electronic mail using z/OS UNIX sendmail,” on page 297 describes how to use z/OS UNIX sendmail, provided with z/OS Communications Server, to prepare and send electronic mail using the facilities of the z/OS shell.
- Chapter 3, “Monitoring the TCP/IP network,” on page 303 describes how to use the following TCP/IP commands to obtain information from the network:

- The TSO NETSTAT and z/OS UNIX **netstat** commands
  - The TSO PING and z/OS UNIX **ping** commands
  - The TSO RPCINFO and z/OS UNIX **rpcinfo**/**orpcinfo** commands
  - The TSO TRACERTE and z/OS UNIX **traceroute** commands
- Chapter 4, "Managing network security," on page 695 describes how to use the following commands to obtain or modify security information in the network:
  - The z/OS UNIX **certbundle** command
  - The z/OS UNIX **ipsec** command
  - The z/OS UNIX **nssctl** command
- Chapter 5, "Displaying policy-based networking information," on page 819 describes how to use the z/OS UNIX pasearch command and the z/OS UNIX trmdstat command to display policy based networking information from the network.
- Chapter 6, "Querying and administrating a Domain Name System (DNS)," on page 891 describes the Domain Name System (DNS) domain names, domain name servers, resolvers, and resource records.
- Chapter 7, "Managing TCP/IP network resources with SNMP," on page 953 describes how to use the Simple Network Management Protocol (SNMP) commands and details what support the z/OS Communications Server SNMP agent and subagents provide.
- Chapter 8, "SNTP daemon: Simple Network Time Protocol," on page 995 describes how to use the SNTP daemon.
- Chapter 9, "Browsing and searching syslog daemon files and archives," on page 997
- Appendix A, "SNMP capability statement," on page 1005 includes the SNMP agent and subagents capability statement for z/OS Communications Server.
- Appendix B, "Management Information Base (MIB) objects," on page 1025 lists the objects defined by the Management Information Base (MIB), which are supported by the SNMP agent and subagents on the z/OS Communications Server, and the maximum access allowed.
- Appendix C, "IBM 3172 attribute index," on page 1063 shows the 3172 attributes and their corresponding MIB variables.
- Appendix D, "SNMP trap types," on page 1065 lists the generic and enterprise-specific trap types that can be received by SNMP.
- Appendix E, "ICMP/ICMPv6 types and codes," on page 1071 lists the Internet Control Message Protocol (ICMP) types and codes from *TCP/IP Illustrated, Volume 1 The Protocols*, by W. Richard Stevens.
- Appendix F, "Related protocol specifications," on page 1073 lists the related protocol specifications for TCP/IP.
- "Accessibility," describes accessibility features to help users with physical disabilities.
- "Notices" contains notices and trademarks used in this document.
- "Bibliography" contains descriptions of the documents in the z/OS Communications Server library.

How to use this document

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

Determining whether a publication is current

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

- At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.
- If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.
- To compare softcopy publications, you can check the last 2 characters of the publication's file name (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

How to contact IBM service

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

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

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

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

If you would like to provide feedback on this publication, see “Communicating your comments to IBM” on page 1121.

Using TSO and z/OS UNIX commands in the MVS batch environment

z/OS Communications Server TSO and z/OS UNIX shell commands can be invoked from the MVS batch environment.

TSO commands

For TSO commands, specify a program name IKJEFT01 on your MVS batch JCL EXEC statement. For more information on executing IKJEFT01 in the MVS batch environment, see z/OS TSO/E Customization. For example, to invoke the TSO NETSTAT command with the CONN report option, you could use the following JCL statements:

```
//TSOBATCH JOB MSGCLASS=A
//STEP1    EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSTSIN  DD *
 NETSTAT CONN
//
```

The output from the command is written to the following DD statements:

- SYSTSPRT - Normal command output
- SYSOUT - Error messages

z/OS UNIX shell commands

For z/OS UNIX shell commands, specify a program name BPXPBATCH on your MVS batch JCL EXEC statement. For more information on executing BPXBATCH in the MVS batch environment, see the z/OS UNIX System Services Command Reference. For example, to invoke the z/OS UNIX **netstat** command with the **-c** report option, you could use the following JCL statements:

```
//BPXBATCH JOB
//STEP1    EXEC PGM=BPXBATCH,PARM='SH netstat -c'
//STDOUT DD PATH='/tmp/stdonet',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),PATHMODE=SIRWXU
//STDERR DD PATH='/tmp/stdenet',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),PATHMODE=SIRWXU
//
```

The **netstat** report output is written to z/OS UNIX file /tmp/stdonet.

Conventions and terminology that are used in this document

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

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

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

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

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

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

- RDMA network interface card (RNIC), which is used to refer to the IBM 10GbE RoCE Express® feature.

- Shared RoCE environment, which means that the 10GbE RoCE Express feature operates on an IBM z13™ (z13) or later system, and that the feature can be used concurrently, or shared, by multiple operating system instances. The RoCE Express feature is considered to operate in a shared RoCE environment even if you use it with a single operating system instance.

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

Clarification of notes

Information traditionally qualified as Notes is further qualified as follows:

Note Supplemental detail

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

Guideline
Customary way to perform a procedure

Rule Something you must do; limitations on your actions

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

Requirement
Dependencies, prerequisites

Result Indicates the outcome

How to read a syntax diagram

This syntax information applies to all commands and statements that do not have their own syntax described elsewhere.

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line (the main path).

Symbols and punctuation

The following symbols are used in syntax diagrams:

Symbol
Description

▸▸ Marks the beginning of the command syntax.

▸ Indicates that the command syntax is continued.

| Marks the beginning and end of a fragment or part of the command syntax.

▸◂ Marks the end of the command syntax.

You must include all punctuation such as colons, semicolons, commas, quotation marks, and minus signs that are shown in the syntax diagram.

Commands

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

- When describing how to use the command in a TSO environment, the command is presented in uppercase (for example, NETSTAT).
- When describing how to use the command in a z/OS UNIX environment, the command is presented in bold lowercase (for example, netstat).

Parameters

The following types of parameters are used in syntax diagrams.

Required
Required parameters are displayed on the main path.

Optional
Optional parameters are displayed below the main path.

Default
Default parameters are displayed above the main path.

Parameters are classified as keywords or variables. For the TSO and MVS console commands, the keywords are not case sensitive. You can code them in uppercase or lowercase. If the keyword appears in the syntax diagram in both uppercase and lowercase, the uppercase portion is the abbreviation for the keyword (for example, OPERand).

For the z/OS UNIX commands, the keywords must be entered in the case indicated in the syntax diagram.

Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a data set is a variable.

Syntax examples

In the following example, the PUt subcommand is a keyword. The required variable parameter is *local_file*, and the optional variable parameter is *foreign_file*. Replace the variable parameters with your own values.

```
►►──PUt──local_file──┬──────────────┬──────────────────────────────────►◄
                     └─foreign_file─┘
```

Longer than one line

If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.

```
►►──┤ The first line of a syntax diagram that is longer than one line ├──────────►

►─┤ The continuation of the subcommands, parameters, or both ├───────────────►◄
```

Required operands

Required operands and values appear on the main path line. You must code required operands and values.

```
►►──REQUIRED_OPERAND──────────────────────────────────────────────►◄
```

Optional values

Optional operands and values appear below the main path line. You do not have to code optional operands and values.

```
►►──┬─────────┬───────────────────────────────────────────────────►◄
    └─OPERAND─┘
```

Selecting more than one operand

An arrow returning to the left above a group of operands or values means more than one can be selected, or a single one can be repeated.

```
►►──┬──────────────────────────────────────┬──────────────────────►◄
    │  ┌─,───────────────────────────────┐ │
    │  ▼                                 │ │
    └──┬─REPEATABLE_OPERAND_OR_VALUE_1─┬─┴─┘
       ├─REPEATABLE_OPERAND_OR_VALUE_2─┤
       ├─REPEATABLE_OPER_OR_VALUE_1────┤
       └─REPEATABLE_OPER_OR_VALUE_2────┘
```

Nonalphanumeric characters

If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code OPERAND=(001,0.001).

```
►►──OPERAND──=──(──001──,──0.001──)───────────────────────────────►◄
```

Blank spaces in syntax diagrams

If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code OPERAND=(001 FIXED).

```
►►──OPERAND──=──(──001── ──FIXED──)───────────────────────────────►◄
```

Default operands

Default operands and values appear above the main path line. TCP/IP uses the default if you omit the operand entirely.

```
    ┌─DEFAULT─┐
►►──┤         ├──────────────────────────────────────────────────►◄
    └─OPERAND─┘
```

Variables

A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

```
►►──variable─────────────────────────────────────────────────────►◄
```

Syntax fragments

Some diagrams contain syntax fragments, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

```
►►──┤ Syntax fragment ├──────────────────────────────────────────►◄
```

Syntax fragment:

```
├──1ST_OPERAND──,──2ND_OPERAND──,──3RD_OPERAND───────────────────┤
```

Prerequisite and related information

z/OS Communications Server function is described in the z/OS Communications Server library. Descriptions of those documents are listed in “Bibliography” on page 1111, in the back of this document.

Required information

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

Softcopy information

Softcopy publications are available in the following collection.

Titles	Order Number	Description
IBM System z Redbooks Collection	SK3T-7876	The IBM Redbooks® publications selected for this CD series are taken from the IBM Redbooks inventory of over 800 books. All the Redbooks publications that are of interest to the System z® platform professional are identified by their authors and are included in this collection. The System z subject areas range from e-business application development and enablement to hardware, networking, Linux, solutions, security, parallel sysplex, and many others. For more information about the Redbooks publications, see http://www-03.ibm.com/systems/z/os/zos/zfavorites/.

Other documents

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

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

To find the complete z/OS library, visit the z/OS library in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

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

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

Title	**Number**
DNS and BIND, Fifth Edition, O'Reilly Media, 2006	ISBN 13: 978-0596100575
Routing in the Internet, Second Edition, Christian Huitema (Prentice Hall 1999)	ISBN 13: 978-0130226471
sendmail, Fourth Edition, Bryan Costales, Claus Assmann, George Jansen, and Gregory Shapiro, O'Reilly Media, 2007	ISBN 13: 978-0596510299
SNA Formats	GA27-3136
TCP/IP Illustrated, Volume 1: The Protocols, W. Richard Stevens, Addison-Wesley Professional, 1994	ISBN 13: 978-0201633467
TCP/IP Illustrated, Volume 2: The Implementation, Gary R. Wright and W. Richard Stevens, Addison-Wesley Professional, 1995	ISBN 13: 978-0201633542
TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, W. Richard Stevens, Addison-Wesley Professional, 1996	ISBN 13: 978-0201634952
TCP/IP Tutorial and Technical Overview	GG24-3376
Understanding LDAP	SG24-4986
z/OS Cryptographic Services System SSL Programming	*SC14-7495*
z/OS IBM Tivoli Directory Server Administration and Use for z/OS	*SC23-6788*
z/OS JES2 Initialization and Tuning Guide	*SA32-0991*
z/OS Problem Management	*SC23-6844*
z/OS MVS Diagnosis: Reference	*GA32-0904*
z/OS MVS Diagnosis: Tools and Service Aids	*GA32-0905*
z/OS MVS Using the Subsystem Interface	*SA38-0679*
z/OS V2R1 Program Directory	*GI11-9848*
z/OS UNIX System Services Command Reference	*SA23-2280*
z/OS UNIX System Services Planning	*GA32-0884*
z/OS UNIX System Services Programming: Assembler Callable Services Reference	*SA23-2281*
z/OS UNIX System Services User's Guide	*SA23-2279*
z/OS XL C/C++ Runtime Library Reference	*SC14-7314*
zEnterprise System and System z10 OSA-Express Customer's Guide and Reference	*SA22-7935*

Redbooks publications

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

Title	Number
IBM z/OS V2R1 Communications Server TCP/IP Implementation, Volume 1: Base Functions, Connectivity, and Routing	SG24-8096
IBM z/OS V2R1 Communications Server TCP/IP Implementation, Volume 2: Standard Applications	SG24-8097
IBM z/OS V2R1 Communications Server TCP/IP Implementation, Volume 3: High Availability, Scalability, and Performance	SG24-8098
IBM z/OS V2R1 Communications Server TCP/IP Implementation, Volume 4: Security and Policy-Based Networking	SG24-8099
IBM Communication Controller Migration Guide	SG24-6298
IP Network Design Guide	SG24-2580
Managing OS/390 TCP/IP with SNMP	SG24-5866
Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender	SG24-5957
SecureWay Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements	SG24-5631
SNA and TCP/IP Integration	SG24-5291
TCP/IP in a Sysplex	SG24-5235
TCP/IP Tutorial and Technical Overview	GG24-3376
Threadsafe Considerations for CICS	SG24-6351

Where to find related information on the Internet

z/OS

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

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

z/OS Internet Library

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

www.ibm.com/systems/z/os/zos/bkserv/

IBM Communications Server product

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

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

IBM Communications Server product support

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

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

IBM Communications Server performance information

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

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

IBM Systems Center publications

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

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

IBM Systems Center flashes

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

http://www.ibm.com/support/techdocs/atsmastr.nsf

Tivoli NetView for z/OS

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

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

RFCs

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

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

Internet drafts

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

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

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

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

DNS websites

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

USENET news groups
comp.protocols.dns.bind

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

BIND Users

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

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

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

The z/OS Basic Skills Information Center

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

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

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS

To access the z/OS Basic Skills Information Center, open your web browser to the following website, which is available to all users (no login required): http://www-01.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zbasics/homepage.html

Summary of changes

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

Changes made in z/OS Version 2 Release 1, as updated February 2015

This document contains information previously presented in z/OS Communications Server: IP System Administrator's Commands, SC27-3661-01, which supported z/OS Version 2 Release 1.

Changed information

- Shared Memory Communications over RDMA adapter (RoCE) virtualization, see "Report field descriptions" on page 445.

Changes made in z/OS Version 2 Release 1, as updated December 2013

This document contains information previously presented in z/OS Communications Server: IP System Administrator's Commands, SC27-3661-00, which supported z/OS Version 2 Release 1.

Changed information

- Network Security Enhancements for SNMP, see "Using the pwtokey facility" on page 973.

Summary of changes for z/OS Version 2 Release 1

For specifics on the enhancements for z/OS Version 2, Release 1, see the following publications:

- z/OS Summary of Message and Interface Changes
- z/OS Introduction and Release Guide
- z/OS Planning for Installation
- z/OS Migration

Chapter 1. Operator commands and system administration

This information describes TSO commands, MVS commands, and related information used to configure TCP/IP and monitor and control the operations of its functions. It is provided as a reference of commonly used commands for experienced system programmers.

MVS commands

After your TCP/IP system is configured, you can use these MVS commands to dynamically start, stop, and control the servers:

- "START command"
- "STOP command" on page 2
- "DISPLAY TCPIP command" on page 2
- "MODIFY command" on page 148
- "VARY TCPIP command" on page 239
- "EZACMD command" on page 285

Recommendation: Although the MVS commands can accept *procname.identifier* to specify the server or address space, the AUTOLOG statement in *hlq*.PROFILE.TCPIP ignores the *identifier* portion. Therefore, it is recommended that you use the member name of the cataloged procedure on the AUTOLOG statements in *hlq*.PROFILE.TCPIP.

START command

Use the START command to dynamically start a TCP/IP server or address space (including the TCP/IP address space). The abbreviated version of the command is the letter S.

```
►►──Start──procname──┬─────────────────────────────────┬──┬──────────────────┬──────────►◄
                     └─,PARMS='(CTRACE(xxxxxxxx))'─────┘  └─,REUSASID=YES────┘
```

procname
: The name of a member in a cataloged procedure library. For the servers, this should be the same name specified on the PORT statement in the PROFILE.TCPIP data set.

,PARMS='(CTRACE(*xxxxxxxx*))'
: Used to start an address space that supports component tracing services (CTRACE). Starts the address space with the specified CTRACE initialization PARMLIB member parameters. Some valid values for *xxxxxxxx* include:

- *CTIRES00* for the Resolver address space
- *CTIEZB00* for the TCP/IP address space
- *CTIORA00* for the OMPROUTE address space

REUSASID=YES
: Specifies that MVS should assign a reusable address space identifier (ASID) to the address space that is being started. This parameter applies only to the following TCP/IP servers and address spaces:

- TCP/IP stack

- Resolver
- TN3270 server

See z/OS MVS Programming: Extended Addressability Guide for more information about reusable ASIDs.

For more information about the Start command, see Start command information in z/OS MVS System Commands.

STOP command

Use the STOP command to stop a TCP/IP server or address space (including the TCP/IP address space) that is in execution. The STOP command can also be used to stop the name server. The abbreviated version of the command is the letter P.

When you issue the STOP command for the TCP/IP address space, one of the following scenarios occurs, depending on whether connected servers have outstanding calls to TCP/IP.

For each server with outstanding calls to TCP/IP

The TCP/IP address space notifies the server that TCP/IP is coming down and requests that the server terminate normally.

If the server does not terminate normally, TCP/IP causes the server to abend with abend code 422. The abend does not appear in a dump; however, it is recorded in the SYS1.LOGREC data set. The outstanding socket call receives error number 1041 EIBMBADPOSTCODE.

For each connected server that does not have outstanding calls

The TCP/IP address space notifies the server that TCP/IP is coming down and drives the server asynchronous error exit routine, if there is one.

```
►►──┬─STOP─┬──procname─────────────────────────────────────►◄
    └─P────┘
```

procname
: The name of the procedure you want to stop. This should be the same member name used to start the server, either on the START command or the AUTOLOG statement in the PROFILE.TCPIP data set.

DISPLAY TCPIP command

Use the DISPLAY TCPIP command from the MVS operator console to display help for a supported command, or to display information received from supported functions. The abbreviated version of the command is the letter D.

The general format of the DISPLAY command is:

```
►►──Display ──TCPIP──,──┬──────────┬──,──┬──────────┬──────────────►◄
                        ├─procname─┤     └─function─┘
                        └─TELNET───┘
```

procname

The name of the member in a procedure library that was used to start the server or address space. You can omit the *procname* parameter when you direct the command to a TCP/IP stack address space and only one TCP/IP stack is currently active. See Table 1 for a list of servers that support the command when a *procname* is specified.

TELNET

Use this parameter to display the name, version, and status of all the TN3270E Telnet servers (Telnet) that are or were running. See "DISPLAY TCPIP,TELNET" on page 145 for more information about this command.

function

Any of the functions that are valid for the server. These functions are described in the following sections.

The following servers or address spaces support the MVS DISPLAY TCPIP command. Not all servers support the same parameters. For further descriptions of the supported parameters, see Table 1.

Table 1. Servers or address spaces that support the MVS DISPLAY TCPIP command

Server or address space	Main parameters	Additional information
TCP/IP address space	HELP, NETSTAT, OMPROUTE, OSAINFO, STOR, SYSPLEX, TRACE	See "DISPLAY command: TCP/IP address space"
TN3270E Telnet server address space	HELP, STOR, TELNET, LUNS, XCF	See "DISPLAY command: TN3270E Telnet server address space" on page 124

DISPLAY TCPIP command examples

```
d tcpip
EZAOP50I TCPIP STATUS REPORT 355
COUNT   TCPIP NAME   VERSION    STATUS
-----   ----------   --------   --------------------------------
    1   TCPCS        CS V1R9    ACTIVE
    2   TCPCS2       CS V1R9    ACTIVE
    3   TCPCS6       CS V1R9    ACTIVE
*** END TCPIP STATUS REPORT ***
EZAOP41I 'DISPLAY TCPIP' COMMAND COMPLETED SUCCESSFULLY
```

DISPLAY command: TCP/IP address space

When you specify a TCP/IP stack name as the *procname* value on the command, you can display information about the TCP/IP stack or about functions that are associated with the stack. For the D TCPIP,,NETSTAT,RESCACHE command, the information is retrieved from the system-wide resolver and is not specific to the TCP/IP stack whose name you specify with the *procname* value.

The functions listed in Table 2 support the DISPLAY TCPIP command when it is directed to a TCP/IP stack address space.

Table 2. Functions that support the DISPLAY TCPIP command in the TCP/IP address space

Function	Command
HELP	"DISPLAY TCPIP,,HELP" on page 4
NETSTAT	"DISPLAY TCPIP,,NETSTAT" on page 9
OMPROUTE	"DISPLAY TCPIP,,OMPROUTE" on page 23

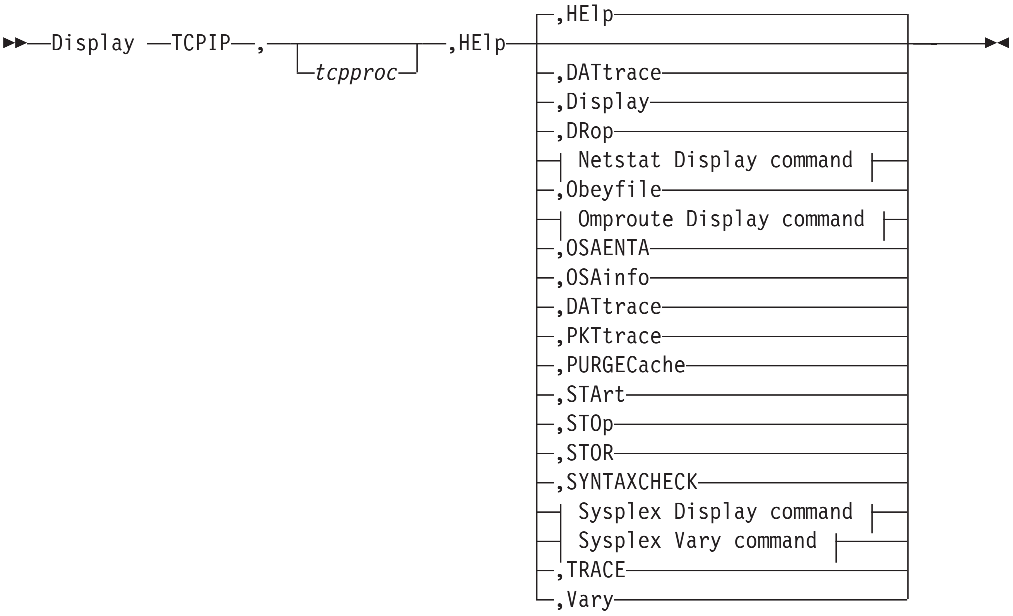
Table 2. Functions that support the DISPLAY TCPIP command in the TCP/IP address space (continued)

Function	Command
OSAINFO	“DISPLAY TCPIP,,OSAINFO” on page 102
STOR	“DISPLAY TCPIP,,STOR” on page 110
SYSPLEX	“DISPLAY TCPIP,,SYSPLEX” on page 111
TRACE	“DISPLAY TCPIP,,TRACE” on page 114

DISPLAY TCPIP,,HELP

Use the DISPLAY TCPIP,HELP command from the MVS operator console to display the syntax of MVS operator commands for TCP/IP.

Format:



Netstat Display command:

```
├──┬─,Netstat──┬──────────────────────────────────────────────────────────┤
   ├─,ACCess───┤
   ├─,ALL──────┤
   ├─,ALLConn──┤
   ├─,ARp──────┤
   ├─,BYTEinfo─┤
   ├─,CACHinfo─┤
   ├─,CONFIG───┤
   ├─,COnn─────┤
   ├─,DEFADDRT─┤
   ├─,DEvlinks─┤
   ├─,DRop─────┤
   ├─,HOme─────┤
   ├─,IDS──────┤
   ├─,ND───────┤
   ├─,PORTList─┤
   ├─,RESCache─┤
   ├─,ROUTe────┤
   ├─,SOCKets──┤
   ├─,SRCIP────┤
   ├─,STATS────┤
   ├─,TTLS─────┤
   ├─,VCRT─────┤
   ├─,VDPT─────┤
   ├─,VIPADCFG─┤
   └─,VIPADyn──┘
```

Omproute Display command:

```
├──┬─,OMProute─┬──────────────────────────────────────────────────────────┤
   ├─,OSPF─────┤
   ├─,RIP──────┤
   ├─,GENERIC──┤
   ├─,RTTABLE──┤
   ├─,IPV6OSPF─┤
   ├─,IPV6RIP──┤
   ├─,GENERIC6─┤
   └─,RT6TABLE─┘
```

Sysplex Display command:

```
├──┬─,SYSplex─┬───────────────────────────────────────────────────────────┤
   ├─,GROUP───┤
   ├─,PORTS───┤
   └─,VIPADyn─┘
```

Sysplex Vary command:

```
├──┬─,SYSplex─────┬───────────────────────────────────────────────────────┤
   ├─,DEACTivate──┤
   ├─,JOINgroup───┤
   ├─,LEAVEgroup──┤
   ├─,QUIesce─────┤
   ├─,REACTivate──┤
   └─,RESUME──────┘
```

Parameters:

Main parameters

HElp
Shows help on the DISPLAY TCPIP,, HELP command.

DATtrace
Shows help on the VARY TCPIP,, DATTRACE command.

Display
Shows help on the DISPLAY TCPIP command.

DRop
Shows help on the VARY TCPIP,, DROP command.

Netstat
Shows help on the DISPLAY TCPIP,, NETSTAT command.

Obeyfile
Shows help on the VARY TCPIP,, OBEYFILE command.

OMProute
Shows help on the DISPLAY TCPIP,, OMPROUTE command.

OSAinfo
Shows help on the DISPLAY TCPIP,, OSAINFO command.

OSAENTA
Shows help on the VARY TCPIP,, OSAENTA command.

PKTtrace
Shows help on the VARY TCPIP,, PKTTRACE command.

PURGECache
Shows help on the VARY TCPIP,, PURGECACHE command.

STArt
Shows help on the VARY TCPIP,, START command.

STOp
Shows help on the VARY TCPIP,, STOP command.

STOR
Shows help on the DISPLAY TCPIP,, STOR command.

SYSplex
Shows help on the DISPLAY TCPIP,, SYSPLEX and VARY TCPIP,, SYSPLEX commands.

TRACE
Shows help on the DISPLAY TCPIP,, TRACE command.

Vary
Shows help on the VARY TCPIP command.

DISPLAY TCPIP,,NETSTAT parameters

ACCess
Shows help on the DISPLAY TCPIP,, NETSTAT, ACCess, NETWORK command.

ALL
Shows help on the DISPLAY TCPIP,, NETSTAT,ALL command.

ALLConn
Shows help on the DISPLAY TCPIP,, NETSTAT,ALLConn command.

ARp
Shows help on the DISPLAY TCPIP,, NETSTAT,ARP command.

BYTEinfo
Shows help on the DISPLAY TCPIP,, NETSTAT,BYTEinfo command.

CACHinfo
Shows help on the DISPLAY TCPIP,, NETSTAT,CACHinfo command.

CONFIG
Shows help on the DISPLAY TCPIP,, NETSTAT,CONFIG command.

COnn
Shows help on the DISPLAY TCPIP,, NETSTAT,COnn commands.

DEFADDRT
Shows help on the DISPLAY TCPIP,, NETSTAT,DEFADDRT command.

DEvlinks
Shows help on the DISPLAY TCPIP,, NETSTAT,DEvlinks command.

HOme
Shows help on the DISPLAY TCPIP,, NETSTAT,HOme command.

ND Shows help on the DISPLAY TCPIP,, NETSTAT,ND command.

IDS
Shows help on the DISPLAY TCPIP,, NETSTAT,IDS command.

PORTList
Shows help on the DISPLAY TCPIP,, NETSTAT,PORTList command.

RESCache
Shows help on the DISPLAY TCPIP,, NETSTAT,RESCache command.

ROUTe
Shows help on the DISPLAY TCPIP,, NETSTAT,ROUTe command.

SOCKets
Shows help on the DISPLAY TCPIP,, NETSTAT,SOCKets command.

SRCIP
Shows help on the DISPLAY TCPIP,, NETSTAT,SRCIP command.

STATS
Shows help on the DISPLAY TCPIP,, NETSTAT,STATS command.

VCRT
Shows help on the DISPLAY TCPIP,, NETSTAT,VCRT command.

TTLS
Shows help on the DISPLAY TCPIP,, NETSTAT,TTLS command.

VDPT
Shows help on the DISPLAY TCPIP,, NETSTAT,VDPT command.

VIPADCFG
Shows help on the DISPLAY TCPIP,, NETSTAT,VIPADCFG command.

VIPADyn
Shows help on the DISPLAY TCPIP,, NETSTAT,VIPADyn and the DISPLAY TCPIP,, SYSPLEX,VIPADyn commands.

DISPLAY TCPIP,,OMPROUTE parameters

OSPF
Shows help on the DISPLAY TCPIP,, OMPROUTE,OSPF command.

RIP
Shows help on the DISPLAY TCPIP,, OMPROUTE,RIP command.

GENERIC
Shows help on the DISPLAY TCPIP,, OMPROUTE,GENERIC command.

RTTABLE
Shows help on the DISPLAY TCPIP,, OMPROUTE,RTTABLE command.

IPV6OSPF
Shows help on the DISPLAY TCPIP,, OMPROUTE,IPV6OSPF command.

IPV6RIP
Shows help on the DISPLAY TCPIP,, OMPROUTE,IPV6RIP command.

GENERIC6
Shows help on the DISPLAY TCPIP,, OMPROUTE,GENERIC6 command.

RT6TABLE
Shows help on the DISPLAY TCPIP,, OMPROUTE,RT6TABLE command.

DISPLAY TCPIP,,SYSPLEX parameters

GROUP
Shows help on the DISPLAY TCPIP,,SYSPLEX,GROUP command.

PORTS
Shows help on the DISPLAY TCPIP,,SYSPLEX,PORTS command.

VIPADyn
Shows help on the DISPLAY TCPIP,,NETSTAT,VIPADyn and the DISPLAY TCPIP,, SYSPLEX,VIPADyn commands.

VARY TCPIP,,SYSPLEX parameters

LEAVEGROUP
Shows help on the VARY TCPIP,, SYSPLEX,LEAVEGROUP command.

JOINgroup
Shows help on the VARY TCPIP,, SYSPLEX,JOINGROUP command.

DEACTIVATE
Shows help on the VARY TCPIP,, SYSPLEX,DEACTIVATE command.

REACTIVATE
Shows help on the VARY TCPIP,, SYSPLEX,REACTIVATE command.

QUIesce
Shows help on the VARY TCPIP,, SYSPLEX,QUIESCE commands.

RESUME
Shows help on the VARY TCPIP,, SYSPLEX,RESUME commands.

Examples:
To view the available help for NETSTAT, issue the following command:

```
d tcpip,,help,netstat
```

```
EZZ0372I D...NETSTAT(,ACCESS|ALL|ALLCONN|ARP|BYTEINFO|CACHINFO|
EZZ0372I CONFIG|CONN|DEFADDRT|DEVLINKS|HOME|IDS|ND|PORTLIST|RESCACHE|ROUTE|
EZZ0372I SOCKETS|SRCIP|STATS|TTLS|VCRT|VDPT|VIPADCFG|VIPADYN)
```

To get more information about the syntax of a particular Netstat command (for example, COnn), issue the following command:

```
d tcpip,,help,conn
```

```
 EZZ0355I D...NETSTAT,CONN<,APPLDATA><,SERVER>
 EZZ0355I <,APPLD=|CLIENT=|CONNTYPE=|IPADDR=|IPPORT=|PORT=|NOTN3270|
 EZZ0355I SMCID=><,FORMAT=LONG|SHORT>
```

To get more information about the syntax of a command (for example, START), issue the following command:

```
d tcpip,tcpa,help,start
EZZ0361I V...(START|CMD=START),XDEVNAME
```

where XDEVNAME is the device name.

DISPLAY TCPIP,,NETSTAT

Use the DISPLAY TCPIP,,NETSTAT command from an operator console to request Netstat information. For a detailed description of each report, see “Netstat report details and examples” on page 324. This command can display only 65 533 lines of output for each report. If the command cannot display all of the report output, the report is truncated and the END OF THE REPORT output line is not displayed. Instead, the following output line is displayed at the end of the report:

```
REPORT TRUNCATED DUE TO GREATER THAN 65533 LINES OF OUTPUT
```

You can use the MAX parameter or filter parameters to limit the number of records that are displayed for a report.

Format:

```
►►──Display ──TCPIP──,──────────────,──────────────────────────────────────────────►
                        └─procname─┘
```

```
►─Netstat,──┬─ACCess,NETWork──┬───────────┬───────────────────────────────────┬──────────────────────►
            │                 └─,ipaddr───┘                                   │
            │              (1) (2) (3) (4) (5) (6) (7)                        │
            ├─ALL──┬────────┬─────────────────────────────────────────────────┤
            │      └─SERVER─┘                                                 │
            │                    (1) (2) (3) (4) (5) (6) (7) (8)              │
            ├─ALLConn──┬─────────────┬────────────────────────────────────────┤
            │          └─,APPLDATA───┘                                        │
            ├─ARp──┬────────────┬─────────────────────────────────────────────┤
            │      └─,netaddr───┘                                             │
            │                         (1) (3) (4)                             │
            ├─BYTEinfo──┬─────────────┬───────────────────────────────────────┤
            │           └─,IDLETIME───┘                                       │
            ├─CACHinfo────────────────────────────────────────────────────────┤
            ├─CONFIG──────────────────────────────────────────────────────────┤
            │       ┌─────────────────┐                                       │
            │       ▼                 │(1) (2) (3) (4) (5) (6) (7) (8)        │
            ├─COnn──────┬─────────────┬───────────────────────────────────────┤
            │           ├─,APPLDATA───┤                                       │
            │           └─,SERVER─────┘                                       │
            ├─DEFADDRT────────────────────────────────────────────────────────┤
            │                  (7) (9)                                        │
            ├─DEvlinks──┬────────┬────────────────────────────────────────────┤
            │           └─,SMC───┘                                            │
            │        (9)                                                      │
            ├─HOme────────────────────────────────────────────────────────────┤
            │                                   (10)                          │
            ├─IDS───┬─────────────────────────────┬───────────────────────────┤
            │       ├─,SUMmary────────────────────┤                           │
            │       └─,PROTOcol=──protocol────────┘                           │
            │      (3)                                                        │
            ├─ND──────────────────────────────────────────────────────────────┤
            │         (2)                                                     │
            ├─PORTList────────────────────────────────────────────────────────┤
            │           ┌─,SUMmary───────────────────┐ (3) (11) (12)          │
            ├─RESCache──┼────────────────────────────┼────────────────────────┤
            │           ├─,DETAIL──┬───────────────┬─┤                        │
            │           │          └─,NEGative─────┘ │                        │
            │           └─,SUMmary───┬─────────┬─────┘                        │
            │                        └─,DNS────┘                              │
            │                  ┌──────────────────────────────┐               │
            │         (3)      ▼                              │               │
            ├─ROUTe───────────────┬───────────────────────────┬───────────────┤
            │                     ├─,ADDRTYPE=──┬─IPV4──┬─────┤               │
            │                     │             └─IPV6──┘     │               │
            │                     ├─,DETAIL───────────────────┤               │
            │                     ├─,IQDIO────────────────────┤               │
            │                     ├─,PR=────┬─ALL─────┬───────┤               │
            │                     │         └─prname──┘       │               │
            │                     ├─,QDIOACCEL────────────────┤               │
            │                     ├─,RADV─────────────────────┤               │
            │                     └─,RSTAT────────────────────┘               │
            │        (1) (2) (3) (4) (5)                                      │
            ├─SOCKets─────────────────────────────────────────────────────────┤
            ├─SRCIP───────────────────────────────────────────────────────────┤
            │       (13)                                                      │
            ├─STATS──────┬──────────────────────────┬─────────────────────────┤
            │            └─,PROTOcol=──protocol─────┘                         │
            │       ┌─,GRoup──────────────────────────┐                       │
            ├─TTLS──┼─────────────────────────────────┼───────────────────────┤
            │       ├─,COnn=connid──┬─────────────┬───┤                       │
            │       │               └─,DETAIL─────┘   │                       │
            │       └─,GRoup───┬─────────────┬────────┘                       │
            │                  └─,DETAIL─────┘                                │
            │                 (2) (3) (5)                                     │
            ├─VCRT──┬─────────────┬───────────────────────────────────────────┤
            │       └─,DETAIL─────┘                                           │
            │                 (2) (3) (5)                                     │
            ├─VDPT──┬─────────────┬───────────────────────────────────────────┤
            │       └─,DETAIL─────┘                                           │
            │                          (3)                                    │
            ├─VIPADCFG──┬─────────────┬───────────────────────────────────────┤
            │           └─,DETAIL─────┘                                       │
            └─VIPADyn──┬───────────────┬──────────────────────────────────────┘
                       ├─,DVIPA────────┤
                       └─,VIPAROUTE────┘
```

```
►──┬──────────────────────────────────────────────┬──────────────────────────────────────►◄
   │                  (6)                         │
   ├─,APPLD=appldata──────────────────────────────┤
   │                  (1)                         │
   ├─,CLIent=──client─────────────────────────────┤
   │                                          (8) │
   ├─,CONNType=──┬─NOTTLSPolicy──────────────┬────┤
   │             └─TTLSPolicy──┬──────────────┬┘    │
   │                           ├─,CURRent─────┤     │
   │                           ├─,GRoup=groupid┤    │
   │                           └─,STALE───────┘     │
   │                      (11)                    │
   ├─,DNSAddr=dnsipaddr───────────────────────────┤
   │                      (12)                    │
   ├─,HOSTName=hostname───────────────────────────┤
   │                         (9)                  │
   ├─,INTFName=──intfname─────────────────────────┤
   │                                   (3)        │
   ├─,IPAddr=──┬─ipaddr───────────────────────────┤
   │           ├─ipaddr/prefixLen───┤             │
   │           └─ipaddr/subnetmask──┘             │
   │                              (5)             │
   ├─,IPPort=─ipaddr+portnum──────────────────────┤
   │              (4)                             │
   ├─,NOTN3270────────────────────────────────────┤
   │                   (2)                        │
   ├─,POrt=──portnum──────────────────────────────┤
   │                     (7)                      │
   └─,SMCID=──┬─smcid─┬───────────────────────────┘
              └─*─────┘
```

```
                                          (14)
├──┬──────────────────────────┬──┬─────────────────┬─────────────────────────────────────┤
   └─,FORMat=──┬─LONG──┬──────┘  ├─,MAX=*──────────┤
               └─SHORT─┘         └─,MAX=──recs─────┘
```

Notes:

1 The CLIent filter is valid only with ALL, ALLConn, BYTEinfo, COnn, and SOCKets.

2 The POrt filter is valid only with ALL, ALLConn, COnn, PORTList, SOCKets, VCRT, and VDPT.

3 The IPAddr filter is valid only with ALL, ALLConn, BYTEinfo, COnn, ND, RESCache, ROUTe, SOCKets, VCRT, VDPT, and VIPADCFG.

4 The NOTN3270 filter is valid only with ALL, ALLConn, BYTEinfo, COnn, and SOCKets.

5 The IPPort filter is valid only with ALL, ALLConn, COnn, SOCKets, VCRT, and VDPT.

6 The APPLD filter is valid only with ALL, ALLConn, and COnn.

7 The SMCID filter is valid only with ALL, ALLConn, COnn, and DEvlinks.

8 The CONNType filter is valid only with ALLConn and COnn.

9 The INTFName filter is valid only with DEvlinks and HOme.

10 The valid protocol values are TCP and UDP.

11 The DNSAddr select string is valid only with RESCache.

12 The HOSTName select string is valid only with RESCache.

13 The valid protocol values are IP, ICMP, TCP, and UDP.

14 If the MAX parameter is not specified on the command, the default value for the MAX parameter is the value of the MAXRECS parameter on the GLOBALCONFIG profile statement.

Parameters:

Note: The minimum abbreviation for each parameter is shown in uppercase letters.

Netstat
Request NETSTAT information.

ACCess,NETWork
Displays information about the network access tree in TCP/IP.

ALL
Displays detailed information about TCP connections and UDP sockets, including some that were recently closed.

SERVER
Provides detailed information only for TCP connections that are in the listen state.

ALLConn
Displays information for all TCP/IP connections, including recently closed ones.

APPLDATA
Displays application data in the output report.

ARp
Displays ARP cache information.

netaddr
This field has a maximum length of 15. Format is *nnn.nnn.nnn.nnn* where *nnn* is in the range 0 - 255. You must code all the triplets. No wildcards are allowed.

BYTEinfo
Displays the byte-count information about each active TCP connection and UDP socket. At the end of the report, the number of records written and the total number of records are displayed. The total number of records represents all UDP sockets and all TCP connections, not just active TCP connections.

IDLETIME
Displays the idle time for each connection.

CACHinfo
Displays information about Fast Response Cache Accelerator statistics. Statistics are displayed for each listening socket configured for Fast Response Cache Accelerator support. There is one section displayed per socket.

CONFIG
Displays TCP/IP configuration data.

COnn
Displays information about each active TCP/IP connection. At the end of the report, the number of records written and the total number of records are displayed. The total number of records represents all UDP sockets and all TCP connections, not just active TCP connections.

APPLDATA
Displays application data in the output report.

SERVER
Displays detailed information about TCP connections in the listen state.

DEFADDRT
Displays the policy table for IPv6 default address selection.

DEvlinks
Displays information about interfaces in the TCP/IP address space.

SMC
Displays only detailed Shared Memory Communications over Remote Direct Memory Access (SMC-R) information about 10GbE RoCE Express interfaces and their associated SMC-R link groups and SMC-R links.

HOme
Displays the home list.

IDS
Displays information about intrusion detection services.

SUMmary
Displays summary information about intrusion detection services.

PROTOcol=*protocol*
Displays information about intrusion detection services for the specified *protocol*. The valid protocols are TCP and UDP.

ND Displays IPv6 Neighbor Discovery cache information.

PORTList
Displays the list of reserved ports and the port access control configuration for unreserved ports. Configure port access control for unreserved ports by specifying PORT profile statements with the port number value replaced by the keyword UNRSV. For more information about port access control, see port access control information in z/OS Communications Server: IP Configuration Guide.

For ports that are reserved by the PORTRANGE profile statement, only one output line is displayed for each range.

RESCache
Displays information about the operation of the system-wide resolver cache. This information is not specific to the TCP/IP stack whose name was specified on the D TCPIP command. Statistical information, such as number of record entries or number of cache queries, can be retrieved, or detailed information about some or all of the cache entries can be retrieved. Resolver caching is configured using resolver configuration statements in the resolver setup file. For more information about resolver caching, see details about resolver caching in z/OS Communications Server: IP Configuration Guide.

DETAIL
Display detailed information for all unexpired entries that are currently in the resolver cache. This information can include the following contents:

- Host-name-to-IP address entries from resolver forward lookups
- IP-address-to-host-name entries from resolver reverse lookups
- Negative entries included in both forward and reverse lookup tables

NEGative
Display detailed information for all negative cache entries in the resolver cache.

SUMmary

Display general system statistics for resolver cache operations. This is the default report for the RESCACHE report option.

DNS

Display general system statistics for resolver cache operations, plus individual statistics for each DNS name server that has provided information that is currently stored in the cache.

Result: Using the DETAIL modifier might cause a large amount of data to be displayed from the MVS console. As an alternative, consider using either the z/OS UNIX shell or TSO version of the command when you have large amount of resolver cache information.

ROUTe

Displays routing information. For a complete description of ROUTe, see "Netstat ROUTe/-r report" on page 524.

Note: Static routes over deleted interfaces are removed from the main routing table and therefore do not appear in the reports generated for the main routing table. Loopback routes are displayed as well as implicit (HOME list) routes.

ADDRTYPE

Displays routing information.

IPV4

Displays IPv4 routing information. This parameter is mutually exclusive with the RADV parameter.

IPV6

Displays IPv6 routing information.

DETAIL

Displays the preceding information plus the metric or cost of use for the route, and displays the following MVS-specific configured parameters for each route:

- Maximum retransmit time
- Minimum retransmit time
- Round-trip gain
- Variance gain
- Variance multiplier

This parameter is mutually exclusive with the QDIOACCEL and IQDIO parameters.

PR

Displays policy-based routing tables. This parameter is mutually exclusive with the QDIOACCEL and IQDIO parameters.

ALL

Displays all policy-based routing tables.

prname

Displays the policy-based routing table that has the name *prname*.

Restriction: Only active policy-based routing tables can be displayed with the Netstat ROUTe command. A policy-based routing table is active if an active routing rule and its associated action reference the policy-based routing table. You can display both active and inactive policy-based routing tables by using the **pasearch** command. For more information, see "The z/OS UNIX pasearch command: Display policies" on page 819.

QDIOACCEL
IQDIO
Displays routes that are eligible for accelerated routing by using the QDIO Accelerator or HiperSockets™ Accelerator. See information about QDIO Accelerator and efficient routing using HiperSockets Accelerator in z/OS Communications Server: IP Configuration Guide for more details. This parameter is mutually exclusive with the DETAIL, PR, RADV, and RSTAT parameters.

RADV
Displays all of the IPv6 routes that are added based on information received in router advertisement messages. All IPv6 router advertisement routes are displayed regardless of whether they are currently used for routing. The flags and reference count are not displayed on the report. This parameter is mutually exclusive with the RSTAT, QDIOACCEL, IQDIO, and ADDRTYPE=IPV4 parameters.

RSTAT
Displays all of the static routes that are defined as replaceable. All defined replaceable static routes are displayed without regard to whether they are currently being used for routing. The flags and reference count are not displayed on the report. The MTU value that is displayed in this report is the value that was defined by using the MTU parameter in the ROUTE statement, or the default value for the specified interface type. This parameter is mutually exclusive with the RADV, QDIOACCEL, and IQDIO parameters.

SOCKets
Displays information for open TCP or UDP sockets that are associated with a client name.

SRCIP
Displays information for all job-specific and destination-specific source IP address associations on the TCP/IP address space.

STATS
Displays TCP/IP statistics for each protocol.

PROTOcol=*protocol*
Displays statistics for the specified protocol. The valid protocols are IP, ICMP, TCP, and UDP.

Result: If you specify TCP, you get both TCP and SMC-R statistics.

TTLS
Displays Application Transparent Transport Layer Security (AT-TLS) information for TCP protocol connections.

COnn=*connid*
Displays the name of the AT-TLS policy rule and the names of the associated actions for the specified connection. The specified *connid* is a number assigned by the TCP/IP stack to uniquely identify a socket entity. You can determine the *connid* from the Conn column in the "Netstat ALLConn/-a report" on page 362.

DETAIL
Displays the AT-TLS policy rule and the associated actions for the specified connection.

GRoup

Displays summary information for AT-TLS groups. AT-TLS groups are defined using the TTLSGroupAction policy statement. The AT-TLS group exists as long as the TTLSGroupAction statement is current or as long as there are active connections using the group.

DETAIL

Displays detailed information for AT-TLS groups.

VCRT

Displays the dynamic VIPA Connection Routing Table information.

DETAIL

For each entry that represents an established dynamic VIPA connection or an affinity created by the passive-mode FTP, displays the preceding information plus the policy rule, action information, routing information, and acceleration information.

For each entry that represents an affinity created by the TIMEDAFFINITY parameter on the VIPADISTRIBUTE profile statement, displays the preceding information plus the affinity related information.

VDPT

Displays the dynamic VIPA Destination Port Table information.

DETAIL

If this optional keyword is specified, when the table for TCP/IP stacks is displayed, the output contains policy action information, target responsiveness values, and a Workload Manager weight value (W/Q), on a separate line. If the DETAIL keyword is not specified, the output does not contain this information.

When the table for non-z/OS targets is displayed, the output contains the weight of the non-z/OS target. If the DETAIL keyword is not specified, the output does not contain this information.

VIPADCFG

Displays the current dynamic VIPA configuration information for a host.

VIPADyn

Displays the current dynamic VIPA and VIPAROUTE information for a local host.

DVIPA

Displays the current dynamic VIPA information only.

VIPAROUTE

Displays the current VIPAROUTE information only.

APPLD=*appldata*

Filter the output of the ALL, ALLConn, and COnn reports by using the specified application data appldata. The maximum size for this field is 40 alphanumeric characters.

CLIent=*client*

Specifies a client name that is used to limit the ALL, ALLConn, BYTEinfo, COnn, and SOCKets responses. Maximum size for this field is 8 alphanumeric characters (plus special characters #, $, and @). Wildcards (* and ?) can appear in any position.

CONNType

Specifies a connection type to limit the ALLConn and COnn responses.

NOTTLSPolicy

Displays only those connections that have not been matched to an Application Transparent Transport Layer Security (AT-TLS) rule. This includes connections that were established while the AT-TLS function was disabled (NOTTLS is specified or in effect by default on the TCPCONFIG statement) and all connections that are not using the TCP protocol. For TCP connections that were established while the AT-TLS function was enabled, this includes the following connections:

- Connections for which AT-TLS policy lookup has not yet occurred (typically the first send or receive has not yet been issued).
- Connections for which AT-TLS policy lookup has occurred but for which no matching rule was found.

TTLSPolicy

Displays only connections that match an Application Transparent Transport Layer Security (AT-TLS) rule. This includes only connections that were established while the AT-TLS function was enabled, for which an AT-TLS policy rule was found with the value `TTLSEnabled ON` or `TTLSEnabled OFF` specified in the TTLSGroupAction. Responses can be further limited on AT-TLS connection type. AT-TLS connection type has the following values:

CURRent

Displays only connections that are using AT-TLS where the rule and all actions are still available to be used for new connections.

GRoup=*groupid*

Displays only connections that are using the AT-TLS group specified by the *groupid* value. The specified *groupid* value is a number assigned by the TCP/IP stack to uniquely identify an AT-TLS group. You can determine the *groupid* value from the GroupID field in the Netstat TTLS GROUP report.

STALE

Displays only connections that are using AT-TLS where the rule or at least one action is no longer available to be used for new connections.

DNSAddr=*dnsipaddr*

Filter the output of the RESCache report using the specified DNS IP address *dnsipaddr*.

HOSTName=*hostname*

Filter the output of the RESCache report using the specified host name value *hostname*.

INTFName=*intfname*

Specifies a name that you can use to limit the DEvlinks and HOme report options to a single interface or to a group of interfaces.

For the DEvlinks and HOme report options, the INTFName filter can be one of the following values:

- The link name of a network interface that was configured on a LINK profile statement (this option selects one interface).
- The interface name of a network interface that was configured on an INTERFACE profile statement (this option selects one interface).
- The port name of an OSA-Express feature in QDIO mode. This is the name that is specified on the PORTNAME keyword in the TRLE (this option selects all interfaces that are associated with the OSA-Express port, including an OSAENTA trace interface).

- The name of a HiperSockets TRLE (this option selects all interfaces that are associated with the HiperSockets TRLE).

Additionally, for the DEvlinks report option, the INTFName filter can also be the interface name of an OSAENTA trace interface, which is EZANTA*portname*, where the *portname* value is the name that is specified on the PORTNAME keyword in the TRLE for the OSA-Express port that is being traced (this option selects one interface).

IPAddr

Provides the option response on specified *ipaddr*, *ipaddr/subnetmask* or *ipaddr/prefixlength*

ipaddr Provides the response for ALL, ALLConn, BYTEinfo, COnn, ND, RESCache, ROUTe, SOCKets, VCRT, and VDPT on the specified IP address (*ipaddr*). Except for the RESCache option, with IPv4 addresses, the default subnet mask 255.255.255.255 is used; for IPv6 addresses, the default prefix length 128 is used. The RECache option does not support any default subnet mask or default prefix length.

ipaddr/subnetmask

Provides the response for ALL, ALLConn, BYTEinfo, COnn, ROUTe, SOCKets, VCRT, and VDPT on the specified IP address with specified subnet mask (*ipaddr/subnetmask*). The IP address (*ipaddr*) in this format must be an IPv4 IP address.

ipaddr/prefixlength

Provides the response for ALL, ALLConn, BYTEinfo, COnn, ND, ROUTe, SOCKets, VCRT, and VDPT on the specified IP address and prefix length. For IPv4 addresses, the prefix length range is 1 - 32. For IPv6 addresses, the prefix length range is 1 - 128.

IPPort=*ipaddr+portnum*

Specifies the IP address and port that are used to limit the ALL, ALLConn, COnn, SOCKets, VCRT, and VDPT report options to the TCP local endpoints, TCP remote endpoints, or the UDP local endpoint. The specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; the specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. For TCP, the filter values *ipaddr* and *portnum* match any combination of the local and remote IP address and local and remote port.

NOTN3270

Provides the response of ALL, ALLConn, BYTEinfo, COnn, and SOCKets, excluding TN3270E Telnet server connections.

POrt=*portnum*

Specifies a port that is used to limit the ALL, ALLConn, COnn, PORTList, SOCKets, VCRT, and VDPT options. The port value range, for all options except the PORTLIST option, is 0 - 65535. No wildcards are allowed. For the PORTList option only, the port value range is 1 - 65535 and you can also filter on the keyword UNRSV.

SMCID=*smcid*

Specifies a Shared Memory Communications - RDMA (SMC-R) link or link group identifier that is used to limit the ALL, ALLConn, COnn, and DEvlinks report options. If an asterisk (*) is specified for the filter value, Netstat provides output only for entries that are associated with SMC-R link, and link groups.

MAX=*recs*

The maximum number of records for which Netstat displays information on the console. The value *recs* indicates the number of records that are displayed on each report. For example, for the connection-related reports, a record is a TCP connection or listener, or a UDP endpoint. Valid *recs* values are in the range 1 - 65535. Specify an asterisk (*) to display information for all records on the console. If the number of output lines exceeds the maximum number of lines for a multi-line WTO (Write to Operator) message, the report output is truncated.

This parameter applies to the ACCess, ALL, ALLConn, ARp, BYTEinfo, CACHinfo, COnn, DEFADDRT, DEvlinks, HOme, IDS, ND, PORTList, RESCache, ROUTe, SOCKets, SRCIP, VCRT, VDPT, and VIPADyn reports. The following list shows the descriptions of variations in support for the parameter for specific reports:

- DEvlinks report - The parameter and the values in the `n OF m RECORDS DISPLAYED` output line apply only to network interfaces that are defined with DEVICE or INTERFACE profile statements. These parameters and values do not apply to the LAN group or to the OSA-Express network traffic analyzer information.
- HOme - The parameter and the values in the `n OF m RECORDS DISPLAYED` output line apply to the IP addresses that are displayed by the report.

If this parameter is specified, it overrides the MAXRECS parameter value on the GLOBALCONFIG profile statement. If this parameter is not specified, the number of records value used for the report is one of the following vlaues:

- The MAXRECS parameter value that is specified on the GLOBALCONFIG TCP/IP profile statement.
- If the MAXRECS parameter is not specified, the MAXRECS parameter default value of 100 records.

The number of records that are displayed and the total number of records that could have been displayed are listed at the end of the report in the following output line, where *n* is the number of records that are displayed and *m* is the total number of records that could be displayed.

```
n OF m RECORDS DISPLAYED
```

If the report output is truncated, the *n* value specifies the number of records for which all output lines are successfully displayed.

Examples:

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report:
Use the DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK[,*ipaddr*] command to display the current NETACCESS profile statement configuration and associated security product information. When you specify the optional *ipaddr* value, the report is limited to the single NETACCESS entry, if any, that is currently being used by the stack for the specified IP address.

Parameters:

ipaddr
A fully qualified IPv4 or IPv6 IP address. Wildcard IP address values are not supported. This value is used to display the NETACCESS profile statement entry that governs the specified *ipaddr* value.

Examples:
Not IPv6 enabled (SHORT format):

```
 NETWORK ACCESS INFORMATION
 INBOUND: YES  OUTBOUND: YES      CACHE: ALL
 NETWORK PREFIX  ADDRESS MASK     SAF NAME
 DEFAULTHOME     <NONE>           DEFLTHOM
   PRFNM: EZB.NETACCESS.MVS00111.TCPCS100.DEFLTHOM  SECLABEL: SYSMULTI
 DEFAULT         <NONE>           DEFLT
   PRFNM: EZB.NETACCESS.*.*.*                       SECLABEL: OUTSIDER
 10.0.0.0        255.0.0.0        SITENET
   PRFNM: EZB.NETACCESS.*.*.SITE*                   SECLABEL: INTERNAL
 10.240.90.0     255.255.255.224  PAYROLL
   PRFNM: EZB.NETACCESS.*.*.PAYROLL                 SECLABEL: CONFACCT
 10.240.90.32    255.255.255.224  SALES
   PRFNM: EZB.NETACCESS.*.*.SALES                   SECLABEL: <NONE>
 10.240.90.64    255.255.255.224  TRAINING
   PRFNM: <NONE>                                    SECLABEL: <NONE>
 10.240.68.0     255.255.255.0    TESTFLOR
   PRFNM: EZB.NETACCESS.MVS00111.*.TESTFLOR         SECLABEL: SITEEAST
 7 OF  7 RECORDS DISPLAYED
 END OF THE REPORT
```

IPv6 enabled or request for LONG format:

```
NETWORK ACCESS INFORMATION
INBOUND: YES  OUTBOUND: YES  CACHE: ALL
SAF NAME  NETWORK PREFIX AND PREFIX LENGTH
--------  --------------------------------
DEFLTHOM  DEFAULTHOME
  PRFNM: EZB.NETACCESS.MVS00111.TCPCS100.DEFLTHOM  SECLABEL: SYSMULTI
DEFLT     DEFAULT
  PRFNM: EZB.NETACCESS.*.*.*                       SECLABEL: OUTSIDER
SITENET   10.0.0.0/8
  PRFNM: EZB.NETACCESS.*.*.SITE*                   SECLABEL: INTERNAL
PAYROLL   10.240.90.0/27
  PRFNM: EZB.NETACCESS.*.*.PAYROLL*                SECLABEL: CONFACCT
SALES     10.240.90.32/27
  PRFNM: EZB.NETACCESS.*.*.SALES                   SECLABEL: <NONE>
TRAINING  10.240.90.64/27
  PRFNM: <NONE>                                    SECLABEL: <NONE>
TESTFLOR  10.240.68.0/24
  PRFNM: EZB.NETACCESS.MVS00111.*.TESTFLOR         SECLABEL: SITEEAST
SITENET6  2001:0DB8:1::/64
  PRFNM: EZB.NETACCESS.*.*.SITE*                   SECLABEL: INTERNAL
PAYROLL6  2001:0DB8:1:0:9:67:115:66/128
  PRFNM: EZB.NETACCESS.*.*.PAYROLL*                SECLABEL: CONFACCT
7 OF 7 RECORDS DISPLAYED
END OF THE REPORT
```

Report field descriptions:

For a SHORT format report

INBOUND

Indicates whether Network Access Control is active for socket commands associated with inbound processing (accept, bind, and all variants of receive).

Yes Indicates that INBOUND is in effect (the INBOUND parameter was defined in the NETACCESS profile statement).

No Indicates that INBOUND is not in effect (the NOINBOUND parameter was defined or is in effect by default in the NETACCESS profile statement).

OUTBOUND

Indicates whether Network Access Control is active for socket commands associated with outbound processing (connect and all variants of send).

Yes Indicates that OUTBOUND is in effect (the OUTBOUND parameter was defined or is in effect by default in the NETACCESS profile statement).

No Indicates that OUTBOUND is not in effect (the NOOUTBOUND parameter was defined in the NETACCESS profile statement).

CACHE
Indicates the level of caching that is in effect for the Network Access Control access checking.

ALL Indicates that when a SAF call is made to check a user's access to a security zone, the result is cached regardless of whether access is permitted or denied.

PERMIT
Indicates that when a SAF call is made to check a user's access to a security zone, the result is cached when access is permitted, but not when access is denied.

SAME Indicates that when a SAF call is made to check a user's access to a security zone, the result is cached when access is permitted, but not when access is denied. In addition, if the user associated with the socket changes or if the IP address being accessed changes from the previous packet received or sent over the socket, a new SAF call is made for a previously permitted security zone.

SAF NAME
The final qualifier of a security product resource name. The maximum length is eight characters.

PRFNM
The security product profile covering this network security zone resource name. If no profile name covers this resource name or the SERVAUTH resource class is not active, the value `NONE` is displayed.

SECLABEL
The security label configured for the security product profile. If none is configured or the SECLABEL resource class is not active, the value `NONE` is displayed.

NETWORK PREFIX AND ADDRESS MASK
Can be one of the following case:

- The IPv4 IP address configured on a NETACCESS statement entry. It is logically ANDed with the ADDRESS MASK value to create the network address for which access control is required.
- The DEFAULTHOME entry configured on a NETACCESS statement entry. This entry is used for all IP addresses local to this stack that are not covered by a specific entry. This entry does not have an ADDRESS MASK.
- The DEFAULT entry configured on a NETACCESS statement entry. This entry is used for all IP addresses that are not covered by any other entry. This entry does not have an ADDRESS MASK.

For a LONG format report

INBOUND
Indicates whether Network Access Control is active for socket commands associated with inbound processing (accept, bind, and all variants of receive).

Yes Indicates that INBOUND is in effect (the INBOUND parameter was defined in the NETACCESS profile statement),

No Indicates that INBOUND is not in effect (the NOINBOUND parameter was defined or is in effect by default in the NETACCESS profile statement).

OUTBOUND

Indicates whether Network Access Control is active for socket commands associated with outbound processing (connect and all variants of send).

Yes Indicates that OUTBOUND is in effect (the OUTBOUND parameter was defined or is in effect by default in the NETACCESS profile statement).

No Indicates that OUTBOUND is not in effect (the NOOUTBOUND parameter was defined in the NETACCESS profile statement).

CACHE

Indicates the level of caching that is in effect for the Network Access Control access checking.

ALL Indicates that when a SAF call is made to check a user's access to a security zone, the result is cached regardless of whether access is permitted or denied.

PERMIT

Indicates that when a SAF call is made to check a user's access to a security zone, the result is cached when access is permitted, but not when access is denied.

SAME Indicates that when a SAF call is made to check a user's access to a security zone, the result is cached when access is permitted, but not when access is denied. In addition, if the user associated with the socket changes or if the IP address being accessed changes from the previous packet received or sent over the socket, a new SAF call is made for a previously permitted security zone.

SAF NAME

The final qualifier of a security product resource name. The maximum length is eight characters.

NETWORK PREFIX AND PREFIX LENGTH

Can be one of the following case:

- The IPv4 or IPv6 IP address and prefix length configured on a NETACCESS statement entry. (If an IPv4 network mask was configured, the prefix length is derived from it.) The prefix length specifies the left-most number of bits of the IP address to use to create the network address for which access control is required.
- The DEFAULTHOME entry configured on a NETACCESS statement entry. This entry is used for all IP addresses local to this stack that are not covered by a specific entry. This entry does not have a PREFIX LENGTH.
- The DEFAULT entry configured on a NETACCESS statement entry. This entry is used for all IP addresses that are not covered by any other entry. This entry does not have a PREFIX LENGTH.

PRFNM

The security product profile covering this network security zone resource name. If no profile name covers this resource name or the SERVAUTH resource class is not active, the value `NONE` is displayed.

SECLABEL
The security label configured for the security product profile. If none is configured or the SECLABEL resource class is not active, the value NONE is displayed.

DISPLAY TCPIP,,OMPROUTE

Use the DISPLAY TCPIP,,OMPROUTE command to display OMPROUTE configuration and state information.

Format:

```
►►──Display ──TCPIP──,──┬──────────┬──,OMProute──────────────────────────────────►
                        └─procname─┘

►──┬─,OSPF─┤ OSPF options ├────────────────────────────────────────────┬────────►◄
   ├─,RIP─┤ RIP options ├──────────────────────────────────────────────┤
   ├─,GENERIC─┤ GENERIC options ├──────────────────────────────────────┤
   ├─,RTTABLE──┬──────────────────────────┬──┬────────────────────┬────┤
   │           └─,PRtable=──┬─ALL────┬────┘  ├─,DEST=ip_addr──────┤    │
   │                        └─prname─┘       └─,DELETED───────────┘    │
   ├─,IPV6OSPF─┤ IPv6 OSPF options ├───────────────────────────────────┤
   ├─,IPV6RIP─┤ IPv6RIP options ├──────────────────────────────────────┤
   ├─,GENERIC6─┤ GENERIC6 options ├────────────────────────────────────┤
   ├─,RT6TABLE──┬──────────────────────────┬──┬──────────────────────────────┬─┤
   │            └─,PRtable=──┬─ALL────┬────┘  ├─,DEST=──┬─ip_addr───────────┬─┤ │
   │                         └─prname─┘       │         └─ip_addr/prefixlen─┘ │ │
   │                                          └─,DELETED──────────────────────┘ │
   └─,OPTIONS──────────────────────────────────────────────────────────┘
```

OSPF options:

```
├──┬─,LIST──┬─,ALL────────┬──────┬───────────────────────────────────────┤
   │        ├─,AREAS──────┤      │
   │        ├─,InterFaceS─┤      │
   │        ├─,NBMA───────┤      │
   │        ├─,NeighBoRS──┤      │
   │        └─,VLINKS─────┘      │
   ├─┤ LSA command ├─────────────┤
   ├─,AREASUM────────────────────┤
   ├─,EXTERNAL───────────────────┤
   ├─,DATABASE──┬────────────────┬┤
   │            └─,AREAID=area_id┘│
   ├─,DBSIZE─────────────────────┤
   ├─,InterFace──┬──────────────┬┤
   │             └─,NAME=if_name┘│
   ├─,NeighBoR──┬────────────────┬┤
   │            └─,IPADDR=ip_addr┘│
   ├─,ROUTERS────────────────────┤
   └─,STATiStics─────────────────┘
```

LSA command:

```
├──,LSA──,LSTYPE=ls_type──,LSID=lsid──,ORIGinator=ad_router─────────────────►

►──┬─────────────────┬───────────────────────────────────────────────────────┤
   └─,AREAID=area_id─┘
```

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report

RIP options:

```
├──┬─,LIST──┬─,ALL─────────┬─────┬────────────────────────────────────────────┤
   │        ├─,InterFaceS──┤     │
   │        └─,ACCEPTED────┘     │
   ├─,InterFace──┬──────────────┬┤
   │             └─,NAME=if_name─┘│
   └─FILTERS──────────────────────┘
```

GENERIC options:

```
├──┬─,LIST──┬─,ALL─────────┬─┬────────────────────────────────────────────────┤
   │        └─,InterFaceS──┘ │
   └─,InterFace──────────────┘
```

IPv6 OSPF options:

```
├──┬─,ALL─────────────────────────────────────────────┬────────────────────┤
   ├─,AREASUM─────────────────────────────────────────┤
   ├─,InterFace──┬──────────────┬─────────────────────┤
   │             ├─,NAME=if_name─┤                     │
   │             └─,ID=if_id─────┘                     │
   ├─,VLINK──┬────────────────────┬───────────────────┤
   │         └─,ENDPT=router-id───┘                   │
   ├─,NeighBoR──┬───────────────────────────────────┬─┤
   │            └─,ID=router-id──┬─────────────────┬┘ │
   │                             └─,IFNAME=if_name─┘  │
   ├─,DBSIZE──────────────────────────────────────────┤
   ├─┤ IPv6 LSA command ├─────────────────────────────┤
   ├─,EXTERNAL────────────────────────────────────────┤
   ├─,DATABASE──┬──────────────────┬──────────────────┤
   │            └─,AREAID=area_id──┘                  │
   ├─,ROUTERS─────────────────────────────────────────┤
   └─,STATiStics──────────────────────────────────────┘
```

IPv6 LSA command:

```
├──,LSA──,LSTYPE=ls_type──,LSID=lsid──,ORIGinator=ad_router──────────────►

►──┬─────────────────┬──┬─────────────────┬───────────────────────────────┤
   └─,AREAID=area_id─┘  └─,IFNAME=if_name─┘
```

IPv6RIP options:

```
├──┬─,ALL─────────────────────────┬───────────────────────────────────────┤
   ├─,ACCEPTED────────────────────┤
   ├─,InterFace──┬──────────────┬─┤
   │             └─,NAME=if_name─┘ │
   └─,FILTERS─────────────────────┘
```

GENERIC6 options:

```
├──┬─,ALL──────────────────────┬──────────────────────────────────────────────┤
   └─,InterFace─┬──────────────┬┘
                └─,NAME=if_name─┘
```

Parameters:

procname
The name of the member in a procedure library that was used to start the associated TCP/IP stack.

OPTIONS
Specifies that the global configuration options information is to be displayed.

OSPF
Specifies that OSPF information is to be displayed.

LIST
Specifies that OSPF information is to be displayed as defined in the OMPROUTE configuration file.

ALL
Displays a comprehensive list of all configuration information.

AREAS
Displays all information concerning configured OSPF areas and their associated ranges.

InterFaceS
Displays, for each OSPF interface, the IP address and configured parameters as coded in the OMPROUTE configuration file.

NBMA
Displays the interface address and polling interval related to interfaces connected to non-broadcast multiaccess networks.

NeighBoRS
Displays the configured neighbors on non-broadcast networks.

VLINKS
Displays all virtual links that have been configured with this router as an endpoint.

LSA
Displays the contents of a single link state advertisement contained in the OSPF database.

A link state advertisement is defined by its

- Link state type (**LSTYPE=***ls_type*)
- Link state ID (**LSID=***lsid*)
- Advertising router (**ORIGinator=***ad_router*)

There is also a separate link state database for each OSPF area. **AREAID=***area_id* on the command line tells the software which database you want to search. If you do not specify which area to search, the backbone (0.0.0.0) area is searched. The different kinds of advertisements, which depend on the value given for link-state-type, are:

Router links (LSTYPE=1)
Describe the set of interfaces attached to a router.

Network links (LSTYPE=2)
Describe the set of routers attached to a network.

Summary link, IP network (LSTYPE=3)
Describe interarea routes to networks.

Summary link, ASBR (LSTYPE=4)
Describe interarea routes to AS boundary routers.

AS external link (LSTYPE=5)
Describe routes to destinations external to the Autonomous System.

Note: The `ORIGINATOR` value must be specified only for link-state-types 3, 4, and 5. An `AREAID` value must be specified for link-state-types 1-4.

Link State IDs, originators (specified by their router IDs), and area IDs take the same format as IP addresses. For example, the backbone area would be entered as `0.0.0.0`

`AREASUM`
Displays the statistics and parameters for all OSPF areas that are attached to the router.

`EXTERNAL`
Displays the AS external advertisements belonging to the OSPF routing domain. One line is printed for each advertisement.

`DATABASE,AREAID=`*`area_id`*
Displays a description of the contents of a particular OSPF area link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. If an AREAID value is not specified, the database from area 0.0.0.0 is displayed.

`DBSIZE`
Displays the number of link state advertisements that are currently in the link state database, categorized by type

`InterFace,NAME=`*`if_name`*
Displays current run-time statistics and parameters related to OSPF interfaces. If a `NAME=`*`if_name`* parameter is omitted, a single line is printed that summarizes each interface. If a `NAME=`*`if_name`* parameter is specified, detailed statistics for that interface are displayed.

`NeighBoR,IPADDR=`*`ip_addr`*
Displays the statistics and parameters that are related to OSPF neighbors. If an `IPADDR=`*`ip_addr`* parameter is omitted, a single line is printed that summarizes each neighbor. If an `IPADDR=`*`ip_addr`* parameter is given, detailed statistics for that neighbor are displayed.

`ROUTERS`
Displays all routes to area-border routers and autonomous system boundary routers that have been calculated by OSPF and are currently present in the routing table.

`STATiStics`
Displays statistics generated by the OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization. Many of the displayed fields are confirmation of the OSPF configuration.

`RIP`
Specifies that RIP information is to be displayed.

LIST

Specifies that RIP information is to be displayed as defined in the OMPROUTE configuration file.

ALL

Display all RIP-related configuration information.

InterFaceS

Display IP addresses and configured parameters for each RIP interface.

ACCEPTED

Displays the routes to be unconditionally accepted, as configured with the ACCEPT_RIP_ROUTE statement.

InterFace,NAME=*if_name*

Displays statistics and parameters related to RIP interfaces. If a NAME=*if_name* parameter is omitted, a single line is printed that summarizes each interface. If a NAME=*if_name* parameter is given, detailed statistics for the specified interface (*if_name*) are displayed.

FILTERS

Displays the global RIP filters.

GENERIC

Specifies that IPv4 information not related to a specific routing protocol is to be displayed.

LIST

Specifies that information is to be displayed as defined in the OMPROUTE configuration file.

ALL

Displays all IPv4 information that is not related to a specific routing protocol.

InterFaceS

Lists all generic IPv4 interfaces that are defined to OMPROUTE using INTERFACE statements.

InterFace

Displays statistics and parameters related to IPv4 generic interfaces that are known to TCP/IP.

RTTABLE

Displays routes in an OMPROUTE IPv4 routing table. If the DISPLAY TCPIP,,OMPROUTE command is issued without the PRtable option, routes from the main routing table are displayed.

DEST=*ip_addr*

Displays the routes to a particular destination. When multiple equal-cost routes exist, use this option to obtain a list of the next hops. You cannot use this option with the DELETED option.

PRtable=ALL

Displays routes in all of the OMPROUTE IPv4 policy-based routing tables. The dynamic routing parameters configured to the Policy Agent for a table are displayed following the routes for the table.

PRtable=*prname*

Displays routes in the specified OMPROUTE IPv4 policy-based routing table. The dynamic routing parameters that are configured to the Policy Agent for the table are displayed following the routes for the table.

DELETED
Displays information about routes that have been deleted from the OMPROUTE routing table and that have not been replaced. You cannot use this option with the DEST=*ip_addr* option.

Results:

- If the RIP protocol is running, deleted routes are displayable for only 3 minutes after deletion. After 3 minutes have elapsed, they become undisplayable.
- If a policy-based routing table is configured to the Policy Agent with no dynamic routing parameters, OMPROUTE has no knowledge of that route table. The route table does not appear in the display of OMPROUTE route tables.
- Only active policy-based routing tables appear in the display of OMPROUTE route tables. A policy-based routing table is active if it is referenced by an active routing rule and its associated action.
- The RTTABLE parameter displays the contents of the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.

IPV6OSPF
Specifies that IPv6 OSPF information is to be displayed.

ALL
Displays a comprehensive list of IPv6 OSPF information.

AREASUM
Displays the statistics and parameters for all IPv6 OSPF areas attached to the router.

InterFace,NAME=*if_name* or InterFace,ID=*if_id*
Displays current run-time statistics and parameters related to IPv6 OSPF interfaces. If the NAME= and ID= parameters are omitted, a single line is printed that summarizes each interface. If the NAME= or ID= parameter is specified, detailed statistics for that interface are displayed.

VLINK,ENDPT=*router-id*
Displays current run-time statistics and parameters related to IPv6 OSPF virtual links. If the ENDPT= parameter is omitted, a single line is printed that summarizes each virtual link. If the ENDPT= parameter is specified, detailed statistics for that virtual link are displayed.

NeighBoR,ID=*router-id*,IFNAME=*if_name*
Displays the statistics and parameters related to IPv6 OSPF neighbors.

- If the ID= parameter is omitted, a single line is printed that summarizes each neighbor.
- If the ID= parameter is given, detailed statistics for that neighbor are displayed.
- If the neighbor specified by the ID= parameter has more than one neighbor relationship with OMPROUTE (for example if there are multiple IPv6 OSPF links connecting them), the IFNAME= parameter can be used to specify which link's adjacency to examine (for an adjacency over a virtual link, specify IFNAME=*).

DBSIZE
Displays the number of link state advertisements that are currently in the IPv6 OSPF link state database, categorized by type.

LSA
Displays the contents of a single link state advertisement contained in the IPv6 OSPF database. A link state advertisement is defined by the following information:

- Link state type (LSTYPE=*ls_type*, where *ls_type* is one of the listed hexadecimal link state type values)
- Link state ID (LSID=*lsid*)
- Advertising router (ORIGinator=*ad_router*)

Each interface has its own set of link LSAs (LSTYPE=0008). IFNAME=*interface_name* on the command line indicates which link's LSA you want to display.

There is also a separate link state database for each IPv6 OSPF area. AREAID=*area_id* on the command line indicates which database you want to search. If you do not specify which area to search, the backbone (0.0.0.0) area is searched. The following list shows different kinds of advertisements, which depend on the value given for link state type:

Router LSA (LSTYPE=2001)
The complete collection describes the state and cost of the router's interfaces to the area. Each router in an area originates one or more Router LSAs.

Network LSA (LSTYPE=2002)
Originated by the designated router of each multiaccess link (for example, LAN) in the area which supports two or more routers. Describes the set of routers that are attached to the link, including the designated router.

Inter-Area Prefix LSA (LSTYPE=2003)
Originated by an area border router. Describes the route to an IPv6 address prefix that belongs to another area.

Inter-Area Router LSA (LSTYPE=2004)
Originated by an area border router. Describes the route to an AS boundary router that belongs to another area.

AS External LSA (LSTYPE=4005)
Originated by an AS boundary router. Describes the route to a destination that is external to the IPv6 OSPF autonomous system.

Link LSA (LSTYPE=0008)
Originated by routers for each link to which they are attached. Provides the router's link-local address, provides a list of IPv6 address prefixes for the link, and asserts a set of options for the network LSA that are originated for the link.

Intra-Area Prefix LSA (LSTYPE=2009)
Originated by routers to advertise one or more IPv6 address prefixes that are associated with the router itself, an attached stub network segment, or an attached transit network segment.

Requirements:

- Specify the AREAID for all link state types except AS External LSA.

Note: If an AREAID value is not specified, the backbone area default value (0.0.0.0) is used.

- Specify the IFNAME value for Link LSAs (LSTYPE=0008).
- Originators (specified by their router IDs) and area IDs are specified in dotted-decimal format. For example, the backbone area is entered as 0.0.0.0.

EXTERNAL
Displays the AS external LSAs belonging to the IPv6 OSPF routing domain. One line is printed for each advertisement.

DATABASE,AREAID=*area_id*
Displays the contents of a particular IPv6 OSPF area link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. If an AREAID value is not specified, the database from area 0.0.0.0 is displayed.

ROUTERS
Displays all routes to other routers that have been calculated by IPv6 OSPF and are currently present in the routing table.

STATISTICS
Displays statistics that are generated by the IPv6 OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization.

IPV6RIP

Specifies that IPv6 RIP information is to be displayed.

ALL
Displays all IPv6 RIP-related information.

ACCEPTED
Displays the routes that are to be unconditionally accepted, as configured with the IPV6_ACCEPT_RIP_ROUTE statement.

InterFace,NAME=*if_name*
Displays statistics and parameters that are related to IPv6 RIP interfaces. If the NAME=*if_name* parameter is omitted, a single line is printed that summarizes each interface. If the NAME=*if_name* parameter is given, detailed statistics for the specified interface (*if_name*) are displayed.

FILTERS
Displays the global IPv6 RIP filters.

GENERIC6
Specifies that IPv6 information not related to a specific dynamic routing protocol is to be displayed.

ALL
Displays all IPv6 information that is not related to a specific routing protocol.

InterFace,NAME=*if_name*
Displays statistics and parameters related to IPv6 generic interfaces that are known to TCP/IP or defined to OMPROUTE with IPV6_INTERFACE statements. If the NAME=*if_name* parameter is omitted, a single line is printed that summarizes each interface. If the NAME=*if_name* parameter is given, detailed statistics for the specified interface (*if_name*) are displayed.

RT6TABLE

Displays all the routes in an OMPROUTE IPv6 routing table. If the **DISPLAY TCPIP,,OMPROUTE** command is issued without the PRtable option, routes from the main routing table are displayed.

DEST=*ip_addr/prefixlen*
Displays information about a particular route. When multiple equal-cost routes exist, use this option to obtain a list of the next hops. You cannot use this option with the DELETED option.

PRtable=ALL
Displays routes in all of the OMPROUTE IPv6 policy-based routing tables. The dynamic routing parameters that are configured to the policy agent for a table are displayed following the routes for the table.

PRtable=*prname*
Displays routes in the specified OMPROUTE IPv6 policy-based routing table. The dynamic routing parameters that are configured to the policy agent for the table are displayed following the routes for the table.

DELETED
Displays information about IPv6 routes that have been deleted from the OMPROUTE routing table and that have not been replaced. You cannot use this option with the DEST=*ip_addr/prefixlen* option.

Results:

- If the IPv6 RIP protocol is running, deleted routes are displayable for only 3 minutes after deletion. After 3 minutes have elapsed, they become undisplayable.
- If a policy-based routing table is configured to the policy agent with no dynamic routing parameters, OMPROUTE has no knowledge of that route table. The routing table is not included in the display of OMPROUTE route tables.
- Only the active policy-based routing tables are included in the display of OMPROUTE route tables. A policy-based routing table is active if an active routing rule and its associated action reference the policy-based routing table.
- The RT6TABLE parameter displays the contents of the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.

Examples:
The following information provides details on the types of data that can be displayed as well as examples of the generated output.

Note: All commands that include the LIST subparameter indicate that the information being displayed is configured information only and does not necessarily mean that the information is actually currently being used by OMPROUTE. To display actual information in current use, use related commands to display current, run-time statistics, and parameters. There are cases when the configured information does not match the actual information that is in use as a result of some undefined or unresolved information in OMPROUTE configuration.

For example, undefined interfaces or parameters in OMPROUTE configuration or incorrect sequence of dynamic reconfiguration using the MODIFY

OMPROUTE,RECONFIG command might result in no update of the actual information. Information that is defined on wildcard interfaces is not displayed in the LIST commands; it is displayed in the corresponding nonLIST commands only when wildcard information is resolved to actual physical interfaces.

Examples using the OPTIONS command:
The DISPLAY TCPIP,*tcpipjobname*,OPTIONS command lists all OMPROUTE global configuration options information. The following contents show a sample output with an explanation of entries:

```
EZZ8172I GLOBAL OPTIONS
    IGNORE UNDEFINED INTERFACES:             YES
```

IGNORE UNDEFINED INTERFACES
Indicates whether the processing of undefined interfaces is ignored.

Examples using the OSPF command:

The following sections show the examples of using the OSPF command.

All OSPF configuration information:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,OSPF,LIST,ALL command lists all OSPF-related configuration information. A sample output with an explanation of entries follows:

```
EZZ7831I GLOBAL CONFIGURATION 967
    TRACE: 2, DEBUG: 4, SADEBUG LEVEL: 0
    STACK AFFINITY:          TCPCS6
    OSPF PROTOCOL:           ENABLED
    EXTERNAL COMPARISON:     TYPE 1
    AS BOUNDARY CAPABILITY:  ENABLED
    IMPORT EXTERNAL ROUTES:  RIP SUB
    ORIG. DEFAULT ROUTE:     ALWAYS
    DEFAULT ROUTE COST:      (1, TYPE 2)
    DEFAULT FORWARD. ADDR:   9.167.100.17
    LEARN HIGHER COST DFLT:  NO
    DEMAND CIRCUITS:         ENABLED
    DR MAX ADJ ATTEMPT:      10

EZZ7832I AREA CONFIGURATION
AREA ID          AUTYPE          STUB? DEFAULT-COST IMPORT-SUMMARIES?
0.0.0.0          0=NONE           NO          N/A           N/A
2.2.2.2          0=NONE           NO          N/A           N/A

--AREA RANGES--
AREA ID          ADDRESS          MASK             ADVERTISE?
2.2.2.2          9.167.200.0      255.255.255.0    YES
2.2.2.2          9.167.100.0      255.255.255.0    YES

EZZ7833I INTERFACE CONFIGURATION
IP ADDRESS      AREA             COST RTRNS TRDLY PRI HELLO DEAD DB_EX
9.169.100.1     0.0.0.0             1   N/A   N/A N/A   N/A  N/A   N/A
9.168.100.3     0.0.0.0             1    10     1   1    20   80   256
9.167.100.13    2.2.2.2             1    10     1   1    20   80   320

                 DEMAND CIRCUIT PARAMETERS
IP ADDRESS        DONOTAGE    HELLO SUPPRESSION   POLL INTERVAL
9.168.100.3       OFF         N/A                   N/A
9.167.100.13      OFF         REQUEST                60

             SUBNET ADVERTISEMENT PARAMETERS
9.168.100.3      9.167.100.13

             ADVERTISED VIPA ROUTES
9.169.100.0   /255.255.255.0    9.169.100.1  /255.255.255.255
```

```
EZZ7836I VIRTUAL LINK CONFIGURATION
 VIRTUAL ENDPOINT TRANSIT AREA  RTRNS TRNSDLY HELLO DEAD DB_EX
 9.67.100.8       2.2.2.2          20     5     40  160   480

EZZ7835I NBMA CONFIGURATION
                INTERFACE ADDR      POLL INTERVAL
                9.168.100.3         120

EZZ7834I NEIGHBOR CONFIGURATION
                NEIGHBOR ADDR     INTERFACE ADDRESS   DR ELIGIBLE?
                9.168.100.56      9.168.100.3         YES
                9.168.100.70      9.168.100.3         NO
```

TRACE
Displays the level of tracing that is currently in use by OMPROUTE for initialization and IPv4 routing protocols.

DEBUG
Displays the level of debugging that is currently in use by OMPROUTE for initialization and IPv4 routing protocols.

SADEBUG LEVEL
Displays the level of debugging that is currently in use by OMPROUTE OSPF SNMP subagent.

STACK AFFINITY
Displays the name of the stack on which OMPROUTE is running.

OSPF PROTOCOL
Indicates whether OSPF is enabled or disabled.

EXTERNAL COMPARISON
Displays the external route type that is used by OSPF when importing external information into the OSPF domain and when comparing OSPF external routes to RIP routes.

AS BOUNDARY CAPABILITY
Indicates whether the router will import external routes into the OSPF domain.

IMPORT EXTERNAL ROUTES
Indicates the types of external routes that are imported into the OSPF domain. Displayed only when AS Boundary Capability is enabled.

ORIG DEFAULT ROUTE
Indicates whether the router will originate a default route into the OSPF domain. The Originate Default Route is displayed only when AS Boundary Capability is enabled.

DEFAULT ROUTE COST
Displays the cost and type of the default route (if advertised). The Default Route Cost is displayed only when AS Boundary Capability is enabled and Orig Default Route value is Always.

DEFAULT FORWARD ADDR
Displays the forwarding address that is specified in the default route (if advertised). The Default Forwarding Address is displayed only when AS Boundary Capability is enabled and Orig Default Route value is Always.

LEARN HIGHER COST DFLT
Indicates the value of the LEARN_DEFAULT_ROUTE parameter of the

AS_BOUNDARY_ROUTING configuration statement. This parameter is displayed only when AS Boundary Capability is enabled and Orig Default Route is Always.

DEMAND CIRCUITS
Indicates whether demand circuit support is available for OSPF interfaces.

DR MAX ADJ ATTEMPT
Specifies a threshold value for maximum number of adjacency attempts to a neighboring designated router. This value is used for reporting and controlling futile neighbor state loops. See the information about futile neighbor state loops in z/OS Communications Server: IP Configuration Guide.

The remainder of the `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,LIST,ALL` output is described as follows:

Configured OSPF areas and ranges:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,LIST,AREAS` command lists all information concerning configured OSPF areas and their associated ranges. A sample output with an explanation of entries follows:

```
EZZ7832I AREA CONFIGURATION 115
AREA ID          AUTYPE          STUB? DEFAULT-COST IMPORT-SUMMARIES?
0.0.0.0          0=NONE           NO          N/A           N/A
2.2.2.2          0=NONE           NO          N/A           N/A

--AREA RANGES--
AREA ID          ADDRESS          MASK             ADVERTISE?
2.2.2.2          9.167.200.0      255.255.255.0    YES
2.2.2.2          9.167.100.0      255.255.255.0    YES
```

AREA ID
Displays the area ID.

AUTYPE
Displays the method used for area authentication. The method *Simple-pass* means that a simple password scheme is being used for the area authentication. The method*MD5* means that MD5 hash is being used for authentication.

STUB?
Indicates whether the area is a stub area.

DEFAULT COST
Displays the cost of the default route that is configured for the stub area.

IMPORT SUMMARIES?
Indicates whether summary advertisements are to be imported into the stub area.

Note: A stub area that does not allow summaries to be imported is sometimes referred to as a totally stubby area.

ADDRESS
Displays the network address for a given range within an area.

MASK
Displays the subnet mask for a given range within an area.

ADVERTISE?
Indicates whether a given range within an area is to be advertised into other areas.

Configured OSPF interfaces:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,LIST,INTERFACES` command lists, for each OSPF interface, the IP address and configured parameters as coded in the OMPROUTE configuration file. (The keyword `IFS` can be substituted for `INTERFACES`.) A sample output with an explanation of entries follows:

```
EZZ7833I INTERFACE CONFIGURATION
IP ADDRESS      AREA             COST RTRNS TRDLY PRI HELLO  DEAD DB_EX
9.168.100.3     0.0.0.0             1    10     1   1    20    80   256
9.167.100.13    2.2.2.2             1    10     1   1    20    80   320
9.169.100.1     0.0.0.0             1   N/A   N/A N/A   N/A   N/A   N/A

                 DEMAND CIRCUIT PARAMETERS
IP ADDRESS        DONOTAGE    HELLO SUPPRESSION   POLL INTERVAL
9.168.100.3       OFF        N/A                    N/A
9.167.100.13      OFF         REQUEST                60

             SUBNET ADVERTISEMENT PARAMETERS
9.168.100.3      9.167.100.13

             ADVERTISED VIPA ROUTES
9.169.100.0   /255.255.255.0    9.169.100.1  /255.255.255.255
```

IP ADDRESS
Indicates the IP address of the interface.

AREA Indicates the OSPF area to which the interface attaches.

COST Indicates the ToS 0 cost (or metric) associated with the interface.

RTRNS
Indicates the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information.

TRDLY
Indicates the transmission delay, which is an estimate of the number of seconds required to transmit routing information over the interface.

PRI Indicates the interface router priority, which is used when selecting the designated router.

HELLO
Indicates the number of seconds between Hello packets sent from the interface.

DEAD
Indicates the number of seconds after not having received an OSPF Hello packet, that a neighbor is declared to be down.

DB_EX
Indicates the number of seconds to allow the database exchange to complete.

DONOTAGE
Indicates whether the interface is configured as a demand circuit.

HELLO SUPPRESSION
Indicates whether the interface is configured for hello suppression.

POLL INTERVAL
Indicates the interval (in seconds) to be used when attempting to contact a neighbor when a neighbor relationship has failed, but the interface is available.

SUBNET ADVERTISEMENT PARAMETERS
Lists the interfaces that are configured with the Subnet parameter

containing a value other than NO. For VIPA interfaces this indicates advertisement of subnet or host routes that are being controlled. For real interfaces this indicates that SUBNET=YES has been coded.

ADVERTISED VIPA ROUTES

Lists the route destinations that OMPROUTE will advertise for locally owned VIPAs. These advertisements are controlled by the Advertise_VIPA_Routes or Subnet parameter on the OSPF_INTERFACE statement.

Configured OSPF nonbroadcast, multiaccess networks:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,LIST,NBMA` command lists the interface address and polling interval related to interfaces connected to non-broadcast multi-access networks. A sample output follows:

```
EZZ7835I NBMA CONFIGURATION 191
                INTERFACE ADDR      POLL INTERVAL
                9.168.100.3         120
```

INTERFACE ADDR

Interface IP address.

POLL INTERVAL

Displays the current poll interval value.

Configured OSPF neighbors:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,LIST,NEIGHBORS` command lists the configured neighbors on non-broadcast networks. (The keyword `NBRS` can be substituted for `NEIGHBORS`.) A sample output with an explanation of entries follows:

```
EZZ7834I NEIGHBOR CONFIGURATION 205
                NEIGHBOR ADDR     INTERFACE ADDRESS   DR ELIGIBLE?
                9.168.100.56      9.168.100.3         YES
                9.168.100.70      9.168.100.3         NO
```

NEIGHBOR ADDR

Indicates the IP address of the neighbor.

INTERFACE ADDRESS

Indicates the IP address of the interface on which the neighbor is configured.

DR ELIGIBLE?

Indicates whether the neighbor is eligible to become the designated router on the link.

Configured OSPF virtual links:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,LIST,VLINKS` command lists all virtual links that have been configured with this router as an endpoint. A sample output with an explanation of entries follows:

```
 EZZ7836I VIRTUAL LINK CONFIGURATION
 VIRTUAL ENDPOINT TRANSIT AREA  RTRNS TRNSDLY HELLO DEAD DB_EX
 9.67.100.8       2.2.2.2          20     5     40  160   480
```

VIRTUAL ENDPOINT

Indicates the OSPF router ID of the other endpoint.

TRANSIT AREA

Indicates the non-backbone area through which the virtual link is configured. Virtual links are treated by the OSPF protocol similarly to point-to-point networks.

RTRNS
Indicates the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information.

TRNSDLY
Indicates the transmission delay, which is an estimate of the number of seconds required to transmit routing information over the interface.

HELLO
Indicates the number of seconds between Hello packets sent from the interface.

DEAD
Indicates the number of seconds after not having received an OSPF Hello packet, that a neighbor is declared to be down.

DB_EX
Indicates the number of seconds to allow the database exchange to complete.

OSPF link state advertisement:
The following command displays the contents of a single link state advertisement contained in the OSPF database:

```
DISPLAY TCPIP,tcpipjobname,OMPROUTE,OSPF,LSA,LSTYPE=ls-type,LSID=lsid,ORIG=ad-router,AREAID
=area-id
```

Tips:

1. For a summary of all the non-external advertisements in the OSPF database, use the following command:
   ```
   DISPLAY TCPIP,tcpipjobname,OMPROUTE,OSPF,DATABASE,AREAID=area-id
   ```
2. For a summary of all the external advertisements in the OSPF database, use the following command:
   ```
   DISPLAY TCPIP,tcpipjobname,OMPROUTE,OSPF,EXTERNAL
   ```

The following example shows an output sample with an explanation of entries:

```
EZZ7880I LSA DETAILS 220
        LS AGE:          292
        LS OPTIONS:      E,DC (0X22)
        LS TYPE:         1
        LS DESTINATION (ID): 9.167.100.13
        LS ORIGINATOR:   9.167.100.13
        LS SEQUENCE NO:  0X80000009
        LS CHECKSUM:     0X8F78
        LS LENGTH:       36
        ROUTER TYPE:  ABR,V (0X05)
        # ROUTER IFCS:   1
              LINK ID:          9.67.100.8
              LINK DATA:        9.167.100.13
              INTERFACE TYPE:   4
                      NO. OF METRICS: 0
                      TOS 0 METRIC:   2 (2)
```

LS AGE
Indicates the age of the advertisement in seconds. An asterisk (*) displayed beside the age value indicates that the originator is supporting demand circuits and has indicated that the LSA should not be aged.

LS OPTIONS
Indicates the optional OSPF capabilities supported by the router that

originated the advertisement. (The value displayed in parentheses is the hexadecimal options value received in the LSA.) These capabilities are denoted by:

LS OPTION	OSPF capability
E	Processes type 5 externals; when this is not set, the area to which the advertisement belongs has been configured as a stub.
T	Can route based on ToS.
MC	RFC 1584 (Multicast Extensions to OSPF) is supported. This value is never set by OMPROUTE but can be received from other routers.
DC	RFC 1793 (Extending OSPF to Support Demand Circuits) is supported.

LS TYPE
Classifies the advertisement and dictates its contents:

LS TYPE	Advertisement
1	Router links advertisement
2	Network link advertisement
3	Summary link advertisement
4	Summary ASBR advertisement
5	AS external link

LS DESTINATION
Identifies what is being described by the advertisement. It depends on the advertisement type. For router links and ASBR summaries, it is the OSPF router ID. For network links, it is the IP address of the network designated router. For summary links and AS external links, it is a network or subnet number.

LS ORIGINATOR
OSPF router ID of the originating router.

LS SEQUENCE NUMBER
Used to distinguish separate instances of the same advertisement. Should be looked at as a signed 32-bit integer. Starts at 0x80000001, and increments by 1 each time the advertisement is updated.

LS CHECKSUM
A checksum of advertisement contents, used to detect data corruption.

LS LENGTH
The size of the advertisement in bytes.

ROUTER TYPE
Indicates the level of function of the advertising router. (The value displayed in parentheses is the hexadecimal router type value received in the LSA).

ROUTER TYPE	Function level
ASBR	The router is an AS boundary router.
ABR	The router is an area border router.
V	The router is an endpoint of an active virtual link that is using the described area as a transit area.

# ROUTER IFCS
The number of router interfaces described in the advertisement.

LINK ID
Indicates what the interface connects to. Depends on interface type. For interfaces to routers (that is, point-to-point links), the Link ID is the neighbor router ID. For interfaces to transit networks, it is the IP address of the network designated router. For interfaces to stub networks, it is the network or subnet number.

LINK DATA
Four bytes of extra information concerning the link; it is either the IP address of the interface (for interfaces to point-to-point networks and transit networks), or the subnet mask (for interfaces to stub networks).

INTERFACE TYPE
One of the following value:

INTERFACE TYPE	Details
1	Point-to-point connection to another router
2	Connection to transit network
3	Connection to stub network
4	Virtual link

NO. OF METRICS
The number of nonzero ToS values for which metrics are provided for this interface. For the z/OS implementation, this value will always be 0.

TOS 0 METRIC
The cost of the interface.

The LS age, LS options, LS type, LS destination, LS originator, LS sequence no, LS checksum and LS length fields are common to all advertisements. The Router type and # router ifcs are seen only in router links advertisements. Each link in the router advertisement is described by the Link ID, Link Data, and Interface type fields.

OSPF area statistics and parameters:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,AREASUM` command displays the statistics and parameters for all OSPF areas attached to the router. A sample output with an explanation of entries follows:

```
EZZ7848I AREA SUMMARY 222
AREA ID        AUTHENTICATION   #IFCS  #NETS  #RTRS  #BRDRS DEMAND
0.0.0.0           NONE              2      0      2      2  ON
2.2.2.2           NONE              1      0      3      2  ON
```

AREA ID
Indicates the ID of the area.

AUTHENTICATION
Indicates the default authentication method for the area.

# IFCS
Indicates the number of router interfaces attached to the particular area. These interfaces are not necessarily functional.

# NETS
Indicates the number of transit networks that have been found while doing the SPF tree calculation for this area.

# RTRS
Indicates the number of routers that have been found when doing the SPF tree calculation for this area.

# BRDRS
Indicates the number of area border routers that have been found when doing the SPF tree calculation for this area.

DEMAND
Indicates whether demand circuits are supported in this area.

OSPF external advertisements:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,EXTERNAL` command lists the AS external advertisements belonging to the OSPF routing domain. One line is printed for each advertisement. Each advertisement is defined by the following three parameters:

- Its link state type (always 5 for AS external advertisements)
- Its link state ID (called the LS destination)
- The advertising router (called the LS originator)

A sample output with an explanation of entries follows:

```
EZZ7853I AREA LINK STATE DATABASE 269
TYPE LS DESTINATION     LS ORIGINATOR     SEQNO     AGE   XSUM
  5 @9.67.100.0         9.67.100.8      0X80000001    4  0X408
  5 @9.169.100.0        9.67.100.8      0X80000001    4  0X73E
  5 @9.169.100.14       9.67.100.8      0X80000001    4  0XE66
  5 @192.8.8.0          9.67.100.8      0X80000001    4  0XAAF
  5 @192.8.8.8          9.67.100.8      0X80000001    4  0X5A4
              # ADVERTISEMENTS:       5
              CHECKSUM TOTAL:         0X2A026
```

TYPE Always 5 for AS external advertisements. An asterisk (*) following the type value indicates that the MC option is on in the advertisement. The MC option indicates that the originating router has implemented RFC 1584 (Multicast Extensions to OSPF). An at sign (@) following the type value indicates that the DC option is on in the advertisement. The DC option indicates that the originating router has implemented RFC 1793 (Extending OSPF to Support Demand Circuits).

LS DESTINATION
Indicates an IP destination (network, subnet, or host). This destination belongs to another Autonomous System.

LS ORIGINATOR
Indicates the router that originated the advertisement.

SEQNO, AGE, and XSUM
It is possible for several instances of an advertisement to be present in the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age), and LS checksum (Xsum) fields are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600. An asterisk (*) displayed beside an age value indicates that the DONOTAGE bit is on.

At the end of the display, the total number of AS external advertisements is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

OSPF area link state database:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,DATABASE,AREAID=`*`area-id`* command displays a description of the contents of a particular OSPF area link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. Each advertisement is defined by the following three parameters:

- Its link state type (called Type)
- Its link state ID (called the LS destination)
- The advertising router (called the LS originator)

A sample output with an explanation of entries follows:

```
EZZ7853I AREA LINK STATE DATABASE 352
TYPE LS DESTINATION     LS ORIGINATOR     SEQNO     AGE   XSUM
  1 @9.67.100.7         9.67.100.7      0X80000016  113  0X5D8D
  1 @9.67.100.8         9.67.100.8      0X80000014   88  0XC0AE
  1 @9.167.100.13       9.167.100.13    0X80000013  100  0X4483
  3 @9.167.100.13       9.167.100.13    0X80000001  760  0XF103
                # ADVERTISEMENTS:       4
                CHECKSUM TOTAL:         0X253C1
```

TYPE Separate LS types are numerically displayed:

TYPE	Description
Type 1	Router links advertisements
Type 2	Network links advertisements
Type 3	Network summaries
Type 4	AS boundary router summaries

An asterisk (*) following the type value indicates that the MC option is on in the advertisement. The MC option indicates that the originating router has implemented RFC 1584 (Multicast Extensions to OSPF). An at sign (@) following the type value indicates that the DC option is on in the advertisement. The DC option indicates that the originating router has implemented RFC 1793 (Extending OSPF to Support Demand Circuits).

LS DESTINATION
Indicates what is being described by the advertisement.

LS ORIGINATOR
Indicates the router that originated the advertisement.

SEQNO, AGE, and XSUM
It is possible for several instances of an advertisement to be present in the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum (Xsum) fields are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600. An asterisk (*) displayed beside an age value indicates that the DONOTAGE bit is on.

At the end of the display, the total number of advertisements in the area database is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement

LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

OSPF link state database statistics:
The `DISPLAY TCPIP,tcpipjobname,OMPROUTE,OSPF,DBSIZE` command displays the number of LSAs currently in the link state database, categorized by type. The following example is a sample output:

```
EZZ7854I LINK STATE DATABASE SIZE 364
                # ROUTER-LSAS:           5
                # NETWORK-LSAS:          0
                # SUMMARY-LSAS:          7
                # SUMMARY ROUTER-LSAS:   1
                # AS EXTERNAL-LSAS:      5
                # INTRA-AREA ROUTES:     4
                # INTER-AREA ROUTES:     0
                # TYPE 1 EXTERNAL ROUTES: 5
                # TYPE 2 EXTERNAL ROUTES: 0
```

OSPF interface statistics and parameters:
The `DISPLAY TCPIP,tcpipjobname,OMPROUTE,OSPF,INTERFACE,NAME=if-name` command displays current, run-time statistics and parameters related to OSPF interfaces. (The keyword `IF` can be substituted for `INTERFACE`.) If no `NAME=` parameter is given (see Example 1), a single line is printed summarizing each interface. If a `NAME=` parameter is given (see Example 2), detailed statistics for that interface are displayed. Sample outputs with an explanation of entries follow:

```
----  Example 1  ----
EZZ7849I INTERFACES 354
IFC ADDRESS     PHYS      ASSOC. AREA    TYPE   STATE    #NBRS    #ADJS
9.168.100.3     CTC1      0.0.0.0         P-P      16        0        0
9.167.100.13    CTC2      2.2.2.2         P-P      16        1        1
10.1.1.1        OSAGBE1   3.3.3.3         BRDCST   32        4        2
10.1.1.2        OSAGBE2   3.3.3.3         BRDCST    2        0        0
0.0.0.0         VL/0      0.0.0.0         VLINK    16        1        1
```

IFC ADDRESS
Interface IP address.

PHYS Displays the interface name.

ASSOC AREA
Attached area ID.

TYPE Interface type. Can be BRDCST (a broadcast interface), P-P (a point-to-point interface), P-2-MP (a point-to-multipoint interface), MULTI (a non-broadcast, multiaccess interface such as ATM), VLINK (an OSPF virtual link), or VIPA (a Virtual IP Address link).

STATE
Can be one of the following value:

STATE	Description
1	Down
1*	Suspend This state is not described in RFC 2328. The interface is suspended because a MODIFY command was issued or because it was unable to establish an adjacency with a neighboring designated router after it exceeded the futile neighbor state loop threshold (DR_Max_Adj_Attempt). For information about futile neighbor state loops, see the futile neighbor state loops information in the z/OS Communications Server: IP Configuration Guide.

STATE	Description
2	Backup
4	Looped back
8	Waiting
16	Point-to-point
32	DR other
64	Backup DR
128	Designated router

For more information about these values, see RFC 1583 (OSPF Version 2).

#NBRS
Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.

#ADJS
Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

```
----  Example 2  ----
non-VIPA interface:
EZZ7850I INTERFACE DETAILS 356
               INTERFACE ADDRESS:      9.168.100.3
               ATTACHED AREA:          0.0.0.0
               PHYSICAL INTERFACE:     CTC1
               INTERFACE MASK:         255.255.255.0
               INTERFACE TYPE:         P-P
               STATE:                  16
               DESIGNATED ROUTER:      N/A
               BACKUP DR:              N/A

DR PRIORITY:     N/A  HELLO INTERVAL:   20  RXMT INTERVAL:    10
DEAD INTERVAL:    80  TX DELAY:          1  POLL INTERVAL:     0
DEMAND CIRCUIT:  OFF  HELLO SUPPRESS:  OFF  SUPPRESS REQ:    OFF
MAX PKT SIZE:    556  TOS 0 COST:        1  DB_EX INTERVAL:  256
 AUTH TYPE: CRYPTO-MD5

# NEIGHBORS:       0  # ADJACENCIES:     0  # FULL ADJS.:      0
# MCAST FLOODS:    0  # MCAST ACKS:      0  # MAX ADJ. RESETS: 0
# ERR PKTS RCVD:   0

NETWORK CAPABILITIES:
 POINT-TO-POINT

VIPA Interface:
EZZ7850I INTERFACE DETAILS 154
                 INTERFACE ADDRESS:      9.67.110.6
                 ATTACHED AREA:          2.2.2.2
                 PHYSICAL INTERFACE:     VIPAIF
                 INTERFACE MASK:         255.255.255.0
                 INTERFACE TYPE:         VIPA
                 STATE:                  32
                 TOS 0 COST:             1
```

INTERFACE ADDRESS
Interface IP address.

ATTACHED AREA
Attached area ID.

PHYSICAL INTERFACE
Displays interface name.

INTERFACE MASK
Displays interface subnet mask.

INTERFACE TYPE
Can be BRDCST (a broadcast interface), P-P (a point-to-point interface), P-2-MP (a point-to-multipoint interface), MULTI (a non-broadcast, multiaccess interface such as ATM), VLINK (an OSPF virtual link), or VIPA (a Virtual IP Address link).

STATE
Can be one of the following value:

STATE	Description
1	Down
1*	Suspend This state is not described in RFC 2328. The interface is suspended because a MODIFY command was issued or because it was unable to establish an adjacency with a neighboring designated router after it exceeded the futile neighbor state loop threshold (DR_Max_Adj_Attempt). For information about futile neighbor state loops, see the futile neighbor state loops information in z/OS Communications Server: IP Configuration Guide.
2	Backup
4	Looped back
8	Waiting
16	Point-to-point
32	DR other
64	Backup DR
128	Designated router

For more information about these values, see RFC 1583 (OSPF Version 2).

DESIGNATED ROUTER
IP address of the designated router.

BACKUP DR
IP address of the backup designated router.

DR PRIORITY
Displays the interface router priority used when selecting the designated router. A higher value indicates that this OMPROUTE is more likely to become the designated router. A value of 0 indicates that OMPROUTE will never become the designated router.

HELLO INTERVAL
Displays the current hello interval value.

RXMT INTERVAL
Displays the current retransmission interval value.

DEAD INTERVAL
Displays the current dead interval value.

TX DELAY
Displays the current transmission delay value.

POLL INTERVAL
Displays the current poll interval value.

DEMAND CIRCUIT
Displays the current demand circuit status.

HELLO SUPPRESS
Displays whether Hello Suppression is currently on or off.

Tip: When a point-to-multipoint interface (displayed Interface type is P-2-MP) on which hello suppression is allowed, an asterisk (*) might be displayed. If an asterisk (*) is displayed, consult the neighbor display for each OSPF neighbor associated with the interface to determine what state of Hello Suppression negotiated with that neighbor.

SUPPRESS REQ
Displays whether Hello Suppression was requested.

MAX PKT SIZE
Displays the maximum size for an OSPF packet sent out this interface.

TOS 0 COST
Displays the interface ToS 0 cost.

DB_EX INTERVAL
Indicates the number of seconds to allow the database exchange to complete.

AUTH TYPE
Authentication type is one of the following value:

NONE
No authentication is used.

Password
Simple password authentication.

MD5 Crypto-MD5 type authentication.

# NEIGHBORS
Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.

# ADJACENCIES
Number of adjacencies. This is the number of neighbors in state Exchange or greater.

# FULL ADJS
Number of full adjacencies. This is the number of neighbors whose state is Full (and therefore with which the router has synchronized databases).

# MAX ADJ. RESETS
Total number of times the maximum threshold value for attempting an adjacency (see the DR MAX ADJ ATTEMPT field) with a neighboring designated router has been reset. The value `N/A` indicates that the field is not applicable for that interface, based on the interface type that is used to reach a neighbor. See futile neighbor state loops information in z/OS Communications Server: IP Configuration Reference for details about the types of interfaces that support futile neighbor state loop detection for OSPF.

# MCAST FLOODS
Number of link state updates that flooded the interface (not counting retransmissions).

# MCAST ACKS
Number of link state acknowledgments that flooded the interface (not counting retransmissions).

# ERR PKTS RCVD
Number of the packets received on the interface that contained errors. These errors include bad packet type, bad length, bad checksum, or other errors.

NETWORK CAPABILITIES
Displays the capabilities of the interface.

OSPF neighbor statistics and parameters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,OSPF,NEIGHBOR,IPADDR=*ip-addr* command displays the statistics and parameters related to OSPF neighbors. (The keyword NBR can be substituted for NEIGHBOR.) If no IPADDR= parameter is given (see Example 1), a single line is printed summarizing each neighbor. If an IPADDR= parameter is given (see Example 2), detailed statistics for that neighbor are displayed. Following are sample outputs with an explanation of entries:

```
----  Example 1  ----
EZZ7851I NEIGHBOR SUMMARY 358
NEIGHBOR ADDR   NEIGHBOR ID     STATE  LSRXL DBSUM LSREQ HSUP IFC
9.167.100.17    9.67.100.7        128     0     0     0  OFF CTC2
VL/0            9.67.100.8        128     0     0     0  OFF *
```

NEIGHBOR ADDR
Displays the neighbor interface IP address.

NEIGHBOR ID
Displays the neighbor OSPF router ID.

STATE
Can be one of the following value:

STATE	Description
1	Down
2	Attempt
4	Init
8	2–Way
16	ExStart
32	Exchange
64	Loading
128	Full

For more information about these values, see RFC 1583 (OSPF Version 2).

LSRXL
Displays the size of the current link state retransmission list for this neighbor.

DBSUM
Displays the size of the database summary list waiting to be sent to the neighbor.

LSREQ
Displays the number of link state advertisements that are being requested from the neighbor.

HSUP Displays whether Hello Suppression is active with the neighbor.

IFC Displays the name of the interface over which a relationship has been established with this neighbor.

```
----  Example 2 ----
EZZ7852I NEIGHBOR DETAILS 360
                NEIGHBOR IP ADDRESS:    9.167.100.17
                OSPF ROUTER ID:         9.67.100.7
                NEIGHBOR STATE:         128
                PHYSICAL INTERFACE:     CTC2
                DR CHOICE:              0.0.0.0
                BACKUP CHOICE:          0.0.0.0
                DR PRIORITY:            1
                NBR OPTIONS:            E,DC (0X22)
 DB SUMM QLEN:      0  LS RXMT QLEN:      0  LS REQ QLEN:       0
 LAST HELLO:        1   NO HELLO:       OFF
 # LS RXMITS:       1  # DIRECT ACKS:     2  # DUP LS RCVD:     2
 # OLD LS RCVD:     0  # DUP ACKS RCVD:   0  # NBR LOSSES:      0
 # ADJ. RESETS:     2  # ERR LS RCVD:     0
```

NEIGHBOR IP ADDRESS

Displays the neighbor interface IP address.

OSPF ROUTER ID

Neighbor OSPF router ID.

NEIGHBOR STATE

Can be one of the following value:

- 1 (Down)
- 2 (Attempt)
- 4 (Init)
- 8 (2-Way)
- 16 (ExStart)
- 32 (Exchange)
- 64 (Loading)
- 128 (Full)

PHYSICAL INTERFACE

Displays the name of the interface over which a relationship has been established with this neighbor.

DR CHOICE, BACKUP CHOICE, DR PRIORITY

Indicates the values seen in the last hello message received from the neighbor.

NBR OPTIONS

Indicates the optional OSPF capabilities supported by the neighbor. (The value displayed in parentheses is the hexadecimal options value received from the neighbor). These capabilities are denoted by:

- E (processes type 5 externals; when this is not set, the area to which the common network belongs has been configured as a stub)
- T (can route based on ToS)
- MC (can forward IP multicast datagrams)
- DC (can support demand circuits)

This field is valid only for those neighbors in state Exchange or greater.

DB SUMM QLEN

Indicates the number of advertisements waiting to be summarized in Database Description packets. It must be 0 except when the neighbor is in state Exchange.

LS RXMT QLEN
Indicates the number of advertisements that have been flooded to the neighbor, but not yet acknowledged.

LS REQ QLEN
Indicates the number of advertisements that are being requested from the neighbor in state Loading.

LAST HELLO
Indicates the number of seconds since a hello message has been received from the neighbor. If the TCP/IP stack enters a storage shortage condition, this value is reset to 0 when the shortage condition is relieved.

NO HELLO
Indicates whether Hello Suppression is active with the neighbor.

# LS RXMITS
Indicates the number of retransmissions that have occurred during flooding.

# DIRECT ACKS
Indicates responses to duplicate link state advertisements.

# DUP LS RCVD
Indicates the number of duplicate retransmissions that have occurred during flooding.

# OLD LS RCVD
Indicates the number of old advertisements received during flooding.

# DUP ACKS RCVD
Indicates the number of duplicate acknowledgments received.

# NBR LOSSES
Indicates the number of times the neighbor has transitioned to Down state.

# ADJ. RESETS
Counts transitions to state ExStart from a higher state.

ERR LS RCVD
Number of the link state advertisements received from the neighbor that are unexpected or that contain errors. These errors include bad advertisement type, bad length, bad checksum, or other errors.

OSPF router routes:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,ROUTERS` command displays all routes to other area-border or autonomous system boundary routers that have been calculated by OSPF and are now present in the routing table. A sample output with an explanation of entries follows:

```
EZZ7855I OSPF ROUTERS 362
DTYPE RTYPE DESTINATION      AREA          COST       NEXT HOP(S)
  BR  SPF   9.67.100.8      2.2.2.2         2         9.167.100.17
  BR  SPF   9.67.100.8      0.0.0.0         2         9.67.100.8
ASBR  SPF   9.67.100.8      2.2.2.2         2         9.167.100.17
```

DTYPE
Indicates the destination type:

ASBR Indicates that the destination is an AS boundary router.

ABR Indicates that the destination is an area border router.

FADD Indicates a forwarding address (for external routes).

RTYPE
Indicates the route type and how the route was derived:

SPF Indicates that the route is an intra-area route (comes from the Dijkstra calculation).

SPIA Indicates that it is an inter-area route (comes from considering summary link advertisements).

DESTINATION
Indicates the destination router OSPF router ID.

AREA Displays the OSPF area to which the destination router belongs.

COST Displays the cost to reach the router.

NEXT HOP(S)
Indicates the address of the next router on the path toward the destination host. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination.

OSPF routing protocol statistics:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,STATISTICS` command displays statistics generated by the OSPF routing protocol. (The keyword `STATS` can be substituted for `STATISTICS`.) The statistics indicate how well the implementation is performing, including its memory and network utilization. Many of the fields displayed are confirmation of the OSPF configuration. The following example shows a sample output with an explanation of entries:

```
EZZ7856I OSPF STATISTICS 380
                OSPF ROUTER ID:         9.167.100.13  (ETH1)
                EXTERNAL COMPARISON:    TYPE 1
                AS BOUNDARY CAPABILITY: YES
                IMPORT EXTERNAL ROUTES: RIP SUB
                ORIG. DEFAULT ROUTE:    ALWAYS
                DEFAULT ROUTE COST:     (1, TYPE2)
                DEFAULT FORWARD. ADDR.: 9.167.100.17
                LEARN HIGHER COST DFLT: NO

ATTACHED AREAS:                2  OSPF PACKETS RCVD:             194
OSPF PACKETS RCVD W/ERRS:      1  TRANSIT NODES ALLOCATED:        82
TRANSIT NODES FREED:          77  LS ADV. ALLOCATED:              53
LS ADV. FREED:                40  QUEUE HEADERS ALLOC:            32
QUEUE HEADERS AVAIL:          32  MAXIMUM LSA SIZE:              512
# DIJKSTRA RUNS:              25  INCREMENTAL SUMM. UPDATES:       0
INCREMENTAL VL UPDATES:        0  MULTICAST PKTS SENT:           227
UNICAST PKTS SENT:            36  LS ADV. AGED OUT:                0
LS ADV. FLUSHED:              10  PTRS TO INVALID LS ADV:          0
INCREMENTAL EXT. UPDATES:     19
```

OSPF ROUTER ID
Displays the router OSPF router ID and its configuration source. Possible sources are:

- OMPROUTE configuration statement (denoted by a prefixed asterisk "*") that has the RouterID parameter specified:
  1. ROUTERID
  2. OSPF
- The name of the IPv4 interface that was used by OMPROUTE to set the router ID. This information is displayed when you do not configure a router ID on an OMPROUTE configuration statement. In this case, the router ID was set by OMPROUTE using the IP address assigned to an IPv4 interface.

For more information about assigned and configured router IDs, see Steps for configuring OSPF and RIP (IPv4 and IPv6) in the z/OS Communications Server: IP Configuration Guide.

EXTERNAL COMPARISON
Displays the external route type used by OSPF when importing external information into the OSPF domain and when comparing OSPF external routes to RIP routes.

AS BOUNDARY CAPABILITY
Displays whether external routes are imported.

IMPORT EXTERNAL ROUTES
Displays the external routes that are imported. Displayed only when AS Boundary Capability is enabled.

ORIG. DEFAULT ROUTE
Displays whether the router will advertise an OSPF default route. Displayed only when AS Boundary Capability is enabled.

DEFAULT ROUTE COST
Displays the cost and type of the default route (if advertised). Displayed only when AS Boundary Capability is enabled and Orig Default Route is ALWAYS.

DEFAULT FORWARD ADDR
Displays the forwarding address specified in the default route (if advertised). Displayed only when AS Boundary Capability is enabled and Orig Default Route is ALWAYS.

LEARN HIGHER COST DFLT
Indicates the value of the LEARN_DEFAULT_ROUTE parameter of the AS_BOUNDARY_ROUTING configuration statement. Displayed only when AS Boundary Capability is enabled and Orig Default Route is ALWAYS.

ATTACHED AREAS
Indicates the number of areas that the router has active interfaces to.

OSPF PACKETS RCVD
Covers all types of OSPF protocol packets.

OSPF PACKETS RCVD W/ERRS
Indicates the number of OSPF packets that have been received that were determined to contain errors.

TRANSIT NODES
Allocated to store router links and network links advertisements.

LS ADV
Allocated to store summary link and AS external link advertisements.

QUEUE HEADERS
Form lists of link state advertisements. These lists are used in the flooding and database exchange processes; if the number of queue headers allocated is not equal to the number available, database synchronization with a neighbor is in progress.

MAXIMUM LSA SIZE
The size of the largest link state advertisement that can be sent.

# DIJKSTRA RUNS
Indicates how many times the OSPF routing table has been calculated from scratch.

INCREMENTAL SUMM UPDATES, INCREMENTAL VL UPDATES
Indicates that new summary link advertisements have caused the routing table to be partially rebuilt.

MULTICAST PKTS SENT
Covers OSPF hello packets and packets sent during the flooding procedure.

UNICAST PKTS SENT
Covers OSPF packet retransmissions and the Database Exchange procedure.

LS ADV. AGED OUT
Indicates the number of advertisements that have hit 60 minutes. Link state advertisements are aged out after 60 minutes. Usually they are refreshed before this time.

LS ADV. FLUSHED
Indicates the number of advertisements removed (and not replaced) from the link state database.

INCREMENTAL EXT. UPDATES
Displays the number of changes to external destinations that are incrementally installed in the routing table.

Examples using the RIP command:

RIP configuration information:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RIP,LIST,ALL` command lists all RIP-related configuration information. A sample output with an explanation of entries follows:

```
EZZ7843I RIP CONFIGURATION 447
TRACE: 1, DEBUG: 0, SADEBUG LEVEL: 0
STACK AFFINITY:  TCPCS6
RIP: ENABLED
RIP DEFAULT ORIGINATION: ALWAYS, COST = 1
PER-INTERFACE ADDRESS FLAGS:
CTC2             9.167.100.13    RIP VERSION 1
                                 SEND NET AND SUBNET ROUTES
                                 RECEIVE NO DYNAMIC HOST ROUTES
                                 RIP INTERFACE INPUT METRIC: 1
                                 RIP INTERFACE OUTPUT METRIC: 0
                                 RIP RECEIVE CONTROL: ANY
CTC1             9.168.100.3     RIP VERSION 1
                                 SEND NET AND SUBNET ROUTES
                                 RECEIVE NO DYNAMIC HOST ROUTES
                                 RIP INTERFACE INPUT METRIC: 1
                                 RIP INTERFACE OUTPUT METRIC: 0
                                 RIP RECEIVE CONTROL: ANY

EZZ7844I RIP ROUTE ACCEPTANCE
ACCEPT RIP UPDATES ALWAYS FOR:
  9.167.100.79         9.167.100.59

IGNORE RIP UPDATES FROM:
NONE
```

TRACE
Displays the level of tracing currently in use by OMPROUTE for initialization and IPv4 routing protocols.

DEBUG
Displays the level of debugging currently in use by OMPROUTE for initialization and IPv4 routing protocols.

SADEBUG LEVEL
Displays the level of debugging currently in use by OMPROUTE OSPF SNMP subagent.

STACK AFFINITY
Displays the name of the stack on which OMPROUTE is running.

The remainder of the `DISPLAY TCPIP,`*tcpipjobname*`,OMPROUTE,RIP,LIST,ALL` output is described in the following sections.

Configured RIP interfaces:
The `DISPLAY TCPIP,`*tcpipjobname*`,OMPROUTE,RIP,LIST,INTERFACES` command lists IP addresses and configured parameters for each RIP interface. (The keyword `IFS` can be substituted for `INTERFACES`.) A sample output with an explanation of entries follows:

```
EZZ7843I RIP CONFIGURATION 447
TRACE: 1, DEBUG: 0, SADEBUG LEVEL: 0
STACK AFFINITY: TCPCS6
RIP: ENABLED
RIP DEFAULT ORIGINATION: ALWAYS, COST = 1
PER-INTERFACE ADDRESS FLAGS:
CTC2            9.167.100.13    RIP VERSION 1
                                SEND NET AND SUBNET ROUTES
                                RECEIVE NO DYNAMIC HOST ROUTES
                                RIP INTERFACE INPUT METRIC: 1
                                RIP INTERFACE OUTPUT METRIC: 0
                                RIP RECEIVE CONTROL: ANY
CTC1            9.168.100.3     RIP VERSION 1
                                SEND NET AND SUBNET ROUTES
                                RECEIVE NO DYNAMIC HOST ROUTES
                                RIP INTERFACE INPUT METRIC: 1
                                RIP INTERFACE OUTPUT METRIC: 0
                                RIP RECEIVE CONTROL: ANY
```

RIP Indicates whether RIP communication is enabled.

RIP DEFAULT ORIGINATION
Indicates the conditions under which RIP supports default route generation and the advertised cost for the default route.

PER-INTERFACE ADDRESS FLAGS
Specifies information about an interface:

RIP VERSION
Specifies whether RIP Version 1 or RIP Version 2 packets are being sent over this interface.

SEND Specifies which types of routes are included in RIP responses sent out on this interface.

RECEIVE
Specifies which types of routes are accepted in RIP responses received on this interface.

RIP INTERFACE INPUT METRIC
Specifies the value of the metric to be added to RIP routes received over this interface.

RIP INTERFACE OUTPUT METRIC
Specifies the value of the metric to be added to RIP routes advertised over this interface.

RIP RECEIVE CONTROL
Indicates what level of RIP updates can be received over the interface. Values are:

ANY RIP1 and RIP2 updates can be received.

NO No RIP updates can be received.

RIP1 Only RIP1 updates can be received.

RIP2 Only RIP2 updates can be received.

RIP routes to be accepted:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RIP,LIST,ACCEPTED` command lists the routes to be unconditionally accepted, as configured with the `ACCEPT_RIP_ROUTE` statement. A sample output follows:

```
EZZ7844I RIP ROUTE ACCEPTANCE
ACCEPT RIP UPDATES ALWAYS FOR:
  9.167.100.79      9.167.100.59
```

ACCEPT RIP UPDATES ALWAYS FOR
Indicates the networks, subnets, and hosts for which updates are always accepted.

RIP interface statistics and parameters:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RIP,INTERFACE,NAME=`*`if-name`* command displays statistics and parameters related to RIP interfaces. (The keyword `IF` can be substituted for `INTERFACE`.) If no `NAME=` parameter is given (`DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RIP,INTERFACE`), a single line is printed summarizing each interface. (See Example 1.) If a `NAME=` parameter is given, detailed statistics for that interface are displayed. (See Example 2.)

```
----  Example 1  ----
EZZ78591 RIP INTERFACES 464
IFC ADDRESS     IFC NAME      SUBNET MASK     MTU    DESTINATION
9.167.100.13    CTC2          255.255.0.0     576    9.167.100.17
```

IFC ADDRESS
Indicates the interface IP address.

IFC NAME
Indicates the interface name.

SUBNET MASK
Indicates the subnet mask.

MTU Indicates the value of the maximum transmission unit.

DESTINATION
Indicates the RIP identification for the destination router when the interface is point-to-point.

```
---- Example 2  ----
EZZ7860I RIP INTERFACE DETAILS 066
INTERFACE ADDRESS:      9.167.100.13
INTERFACE NAME:         CTC2
SUBNET MASK:            255.255.0.0
MTU                     576
DESTINATION ADDRESS:    9.167.100.17

RIP VERSION:            1     SEND POIS. REV. ROUTES: YES
IN METRIC:              1     OUT METRIC:            0
```

```
RECEIVE NET ROUTES:     YES   RECEIVE SUBNET ROUTES:  YES
RECEIVE HOST ROUTES:    NO    SEND DEFAULT ROUTES:    NO
SEND NET ROUTES:        YES   SEND SUBNET ROUTES:     YES
SEND STATIC ROUTES:     NO    SEND HOST ROUTES:       NO

SEND ONLY: VIRTUAL, DEFAULT

FILTERS: SEND            9.67.100.0          255.255.255.0
         RECEIVE         9.67.101.0          255.255.255.0

RIP RECEIVE CONTROL:    ANY
```

INTERFACE ADDRESS
Indicates the interface IP address.

INTERFACE NAME
Indicates the interface name.

SUBNET MASK
Indicates the subnet mask.

MTU Indicates the value of the maximum transmission unit.

DESTINATION ADDRESS
Indicates the RIP identification for the destination router when the interface is point-to-point.

RIP VERSION
Indicates whether RIP Version 1 or RIP Version 2 packets are sent over this interface.

SEND POIS. REV. ROUTES
Indicates whether poisoned reverse routes are advertised in RIP responses sent over this interface. A poisoned reverse route is one with an infinite metric (a metric of 16).

IN METRIC
Specifies the value of the metric to be added to RIP routes received over this interface.

OUT METRIC
Specifies the value of the metric to be added to RIP routes advertised over this interface.

RECEIVE NET ROUTES
Indicates whether network routes are accepted in RIP responses received over this interface.

RECEIVE SUBNET ROUTES
Indicates whether subnet routes are accepted in RIP responses received over this interface.

RECEIVE HOST ROUTES
Indicates whether host routes are accepted in RIP responses received over this interface.

SEND DEFAULT ROUTES
Indicates whether the default route, if available, is advertised in RIP responses sent over this interface.

SEND NET ROUTES
Indicates whether network routes are advertised in RIP responses sent over this interface.

SEND SUBNET ROUTES
Indicates whether subnet routes are advertised in RIP responses sent over this interface.

SEND STATIC ROUTES
Indicates whether static routes are advertised in RIP responses sent over this interface.

SEND HOST ROUTES
Indicates whether host routes are advertised in RIP responses sent over this interface.

SEND ONLY
Indicates the route-type restrictions on RIP broadcasts for this interface.

FILTERS
Indicates the send and receive filters for this interface.

RIP RECEIVE CONTROL
Indicates the type of RIP packets that are received over this interface: RIP1, RIP2, ANY (both RIP1 and RIP2), or NONE.

Global RIP filters:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RIP,FILTERS` command displays the Global RIP filters. A sample output with an explanation of entries follows.

```
EZZ8016I GLOBAL RIP FILTERS
SEND ONLY: VIRTUAL, DEFAULT

IGNORE RIP UPDATES FROM:
  9.67.103.10       9.67.103.9


FILTERS: NOSEND          10.1.1.0            255.255.255.0
         NORECEIVE       9.67.101.0          255.255.255.0
```

SEND ONLY
Indicates the global route-type restrictions on RIP broadcasts that apply to all RIP interfaces.

IGNORE RIP UPDATES FROM
Specifies that RIP routing table broadcasts from this gateway are to be ignored. This option serves as a RIP input filter.

FILTERS
Indicates the global send and receive filters that apply to all RIP interfaces.

Examples using the GENERIC command:

All IPv4 generic information:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,GENERIC,LIST,ALL` command lists all IPv4 configuration information that is not related to a specific routing protocol. A sample output with an explanation of the entries follows:

```
EZZ8053I IPV4 GENERIC CONFIGURATION
TRACE: 2, DEBUG: 3, SADEBUG LEVEL: 0
IPV4 TRACE DESTINATION: /TMP/AMPROUT3.DBG
STACK AFFINITY: TCPCS3

EZZ8056I IPV4 GEN INT CONFIGURATION
IFC NAME         IFC ADDRESS     SUBNET MASK       MTU DESTADDR
NSQDIO3L         9.67.120.3      255.255.255.0     576  N/A
CTC3TO4          9.67.101.3      255.255.255.0   10000  9.67.101.4
```

TRACE
Displays the level of tracing currently in use by OMPROUTE initialization and IPv4 routing protocols.

DEBUG
Displays the level of debugging currently in use by OMPROUTE initialization and IPv4 routing protocols.

SADEBUG LEVEL
Displays the level of debugging currently in use by OMPROUTE OSPF SNMP subagent.

IPV4 TRACE DESTINATION
Indicates the file name of the destination for IPv4 trace, or OMPCTRC if the destination is the OMPROUTE CTRACE.

Restriction: On the console, the file name is shown in upper case, regardless of the case of the actual file name.

STACK AFFINITY
Displays the name of the stack on which OMPROUTE is running.

IPV4 GENERIC INTERFACES
Displays the same output as DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,GENERIC,LIST,INTERFACES described in "Configured IPv4 generic interfaces."

Configured IPv4 generic interfaces:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,GENERIC,LIST,INTERFACES command lists, for each IPv4 generic interface, the IP address and configured parameters that are defined to OMPROUTE using the INTERFACE statement. IFS can be used in place of INTERFACES. A sample output with an explanation of the entries follows:

```
EZZ8056I IPV4 GEN INT CONFIGURATION
IFC NAME         IFC ADDRESS     SUBNET MASK       MTU DESTADDR
NSQDIO3L         9.67.120.3      255.255.255.0     576  N/A
CTC3TO4          9.67.101.3      255.255.255.0   10000  9.67.101.4
```

IFC NAME
The interface link name, as defined using the NAME parameter on the INTERFACE statement.

IFC ADDRESS
The interface home address, as defined using the IP_ADDRESS parameter on the INTERFACE statement.

SUBNET MASK
The interface subnet mask, as defined using the SUBNET_MASK parameter on the INTERFACE statement.

MTU
The interface MTU size, as defined using the MTU parameter on the INTERFACE statement.

DESTADDR
If the interface is known to be a point-to-point interface and the DESTINATION_ADDR parameter was coded in the OMPROUTE configuration file, DESTADDR is the value of the interface DESTINATION_ADDR parameter. Otherwise, N/A is displayed.

IPv4 generic interfaces:
The DISPLAY TCPIP,*tcpname*,OMPROUTE,GENERIC,INTERFACE command displays current, run-time statistics and parameters related to IPv4 generic interfaces that

are known to TCP/IP. The keyword IF can be used instead of INTERFACE. A sample output with an explanation of the entries follows:

```
EZZ8060I IPV4 GENERIC INTERFACES
IFC NAME         IFC ADDRESS     SUBNET MASK       MTU  CFG  IGN
NSQDIO3L         9.67.120.3      255.255.255.0     576  YES   NO
CTC3TO1          130.200.1.3     N/A               N/A   NO  YES
VIPA03           3.3.3.103       N/A               N/A   NO  YES
CTC3TO4          9.67.101.3      255.255.255.0   10000  YES   NO
```

IFC NAME
The interface link name.

IFC ADDRESS
The interface home address.

SUBNET MASK
The interface subnet mask. If the interface is being ignored by OMPROUTE, N/A is displayed.

MTU
The interface MTU size. If the interface is being ignored by OMPROUTE, N/A is displayed.

CFG
Indicates whether or not the interface was configured to OMPROUTE.

IGN
Indicates whether or not the interface is being ignored by OMPROUTE (the value of this field can be YES only if CFG=NO, and the value of GLOBAL_OPTIONS IGNORE_UNDEFINED_INTERFACES is configured to be YES.)

Examples using the RTTABLE command:

OMPROUTE IPv4 main routing table:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE command displays all of the routes in the OMPROUTE IPv4 main routing table. A sample output with an explanation of the entries follows.

Result: This command displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see "DISPLAY TCPIP,,NETSTAT" on page 9.

```
EZZ7847I ROUTING TABLE 796
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   2.0.0.0          FF000000  1       1368    NONE
 SPF   2.2.2.0          FFFFFFFC  3       1380    9.67.106.4
 SPF   2.2.2.2          FFFFFFFF  3       1380    9.67.106.4
SBNT   3.0.0.0          FF000000  1       1549    NONE
 SPF   3.3.3.0          FFFFFFFC  2       1561    9.67.102.3
 SPF   3.3.3.3          FFFFFFFF  2       1561    9.67.102.3
SBNT   4.0.0.0          FF000000  1       1549    NONE
 SPF   4.4.4.4          FFFFFFFC  2       1561    9.67.106.4
 SPF   4.4.4.4          FFFFFFFF  2       1561    9.67.106.4
SBNT   5.0.0.0          FF000000  1       1549    NONE
 SPF   5.5.5.4          FFFFFFFC  2       1567    9.67.107.5
 SPF   5.5.5.5          FFFFFFFF  2       1567    9.67.107.5
SBNT   6.0.0.0          FF000000  1       1549    NONE
 RIP   6.6.6.4          FFFFFFFC  2       30      9.67.103.6
SBNT   7.0.0.0          FF000000  1       1368    NONE
SPIA*  7.7.7.4          FFFFFFFC  3       1380    9.67.106.4
```

```
 DIR*  7.7.7.7          FFFFFFFF  1       1574    VIPA1A
SBNT   8.0.0.0          FF000000  1       1549    NONE
 SPF   8.8.8.8          FFFFFFFC  2       1545    9.67.100.8
 SPF   8.8.8.8          FFFFFFFF  2       1545    9.67.100.8
SBNT   9.0.0.0          FF000000  1       1368    NONE
 DIR*  9.67.100.0       FFFFFF00  1       1576    9.67.100.7
 SPF   9.67.100.7       FFFFFFFF  2       1545    CTC7TO8
 SPF   9.67.100.8       FFFFFFFF  1       1572    9.67.100.8
 SPF   9.67.101.3       FFFFFFFF  2       1561    9.67.106.4
 SPF   9.67.101.4       FFFFFFFF  2       1561    9.67.102.3
 DIR*  9.67.102.0       FFFFFF00  1       1575    9.67.102.7
 SPF   9.67.102.3       FFFFFFFF  1       1566    9.67.102.3
 SPF   9.67.102.7       FFFFFFFF  2       1561    CTC7TO3
 DIR*  9.67.103.0       FFFFFF00  1       1575    9.67.103.7
 RIP   9.67.103.6       FFFFFFFF  1       30      9.67.103.6
 SPF   9.67.105.4       FFFFFFFF  2       1545    9.67.100.8
 SPF   9.67.105.8       FFFFFFFF  2       1561    9.67.106.4
 DIR*  9.67.106.0       FFFFFF00  1       1576    9.67.106.7
 SPF   9.67.106.4       FFFFFFFF  1       1566    9.67.106.4
 SPF   9.67.106.7       FFFFFFFF  2       1561    CTC7TO4
 DIR*  9.67.107.0       FFFFFF00  1       1577    9.67.107.7
 SPF   9.67.107.5       FFFFFFFF  1       1574    9.67.107.5
 SPF   9.67.107.7       FFFFFFFF  2       1566    CTC7TO5
 SPF   9.67.108.2       FFFFFFFF  2       1380    9.67.106.4
 SPF   9.67.108.4       FFFFFFFF  3       1380    9.67.106.4
SBNT   10.0.0.0         FF000000  1       1368    NONE
SPE2   10.1.1.0         FFFFFF00  0       1379    9.67.106.4
SPE2   10.1.1.1         FFFFFFFF  0       1379    9.67.106.4
SBNT   20.0.0.0         FF000000  1       1549    NONE
SPE2   20.1.1.0         FFFFFF00  0       1379    9.67.107.5
SPE2   20.1.1.1         FFFFFFFF  0       1379    9.67.107.5
 RIP   30.0.0.0         FF000000  2       30      9.67.103.6
 RIP   30.1.1.0         FFFFFF00  2       30      9.67.103.6
 RIP % 30.1.1.4         FFFFFFFF  2       30      9.67.103.6
 RIP % 30.1.1.8         FFFFFFFF  2       30      9.67.103.6
SPE2   130.200.0.0      FFFF0000  0       1379    9.67.100.8       (2)
SPE2   130.200.1.1      FFFFFFFF  0       1379    9.67.102.3
SPE2   130.200.1.18     FFFFFFFF  0       1379    9.67.100.8
SPE2   130.201.0.0      FFFF0000  0       1379    9.67.100.8       (2)
SPE2   130.202.0.0      FFFF0000  0       1379    9.67.100.8       (2)
                       0 NETS DELETED, 4 NETS INACTIVE
```

TYPE Indicates how the route was derived:

DFLT Indicates a route defined using the DEFAULT_ROUTE configuration statement in the OMPROUTE configuration file.

SBNT Indicates that the network is subnetted; such an entry is a placeholder only.

DIR Indicates a directly connected network, subnet, or host.

RIP Indicates a route that was learned through the RIP protocol.

DEL Indicates the route has been deleted.

Restriction: Deleted routes are shown in this display only if RIP is active and only as long as RIP needs to advertise to neighboring routers that they have been deleted. Deleted routes cannot be displayed in the detailed routes display.

STAT Indicates a nonreplaceable statically configured route.

SPF Indicates that the route is an OSPF intra-area route.

SPIA Indicates that the route is an OSPF interarea route.

SPE1 Indicates OSPF external routes (type 1).

SPE2 Indicates OSPF external routes (type 2)

RNGE Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.

RSTA Indicates a static route that is defined as replaceable.

An asterisk (*) after the route type indicates that the route has a directly connected backup. A percent sign (%) after the route type indicates that RIP updates are always accepted for this destination.

DEST NET
Indicates the IP destination.

MASK
Indicates the IP destination subnet mask.

COST Indicates the route cost.

Table 3. OMPROUTE IPv4 Route Type and COST Value mapping

Route Type	COST Value
SPF or SPIA	The OSPF cost of the route.
SPE1	The OSPF cost to get to the AS boundary router or forwarding address that is used to reach the destination, plus the external cost.
SPE2	The external cost.
RIP	The RIP metric.
STAT or RSTA	• 0 when the route is direct. • 1 when the route is indirect.
DIR or SBNT	1
RNGE	The OSPF cost of the range.
DFLT	0

AGE Indicates the time that has elapsed since the routing table entry was last refreshed. For routes that have the route type DEL or RIP, this value increments by a factor of 10 for each 10–second increase in age. If the TCP/IP stack enters a storage shortage condition, all routes that have the route type DEL or RIP are refreshed when the shortage condition is relieved.

NEXT HOP(S)
Indicates the IP address of the next router on the path toward the destination. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. Use the `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RTTABLE,DEST=`*`ip-addr`* command to obtain a list of the next hops.

NETS DELETED
Indicates the number of routes that have been deleted from the OMPROUTE routing table and not replaced. Use the D TCPIP,,OMPROUTE,RTTABLE,DELETED command to list these routes.

NETS INACTIVE
Used for internal debugging purposes only.

Route expansion information for the OMPROUTE IPv4 main routing table:
Use the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,DEST=*ip-addr* command to obtain information about a particular route in the OMPROUTE IPv4

main routing table. When multiple equal-cost routes exist, use this command to obtain a list of the next hops. A sample output with an explanation of the entries follows:

Result: This command displays information from the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see "DISPLAY TCPIP,,NETSTAT" on page 9.

```
EZZ7874I ROUTE EXPANSION 370
DESTINATION:    9.68.101.0
MASK:           255.255.255.0
ROUTE TYPE:     SPF
DISTANCE:       6
AGE:            1344
NEXT HOP(S):    9.167.100.17      (CTC2)
                9.168.100.4       (CTC1)
```

DESTINATION
Indicates the IP destination.

MASK
Indicates the IP destination subnet mask.

ROUTE TYPE
Indicates how the route was derived:

DFLT Indicates a route defined using the DEFAULT_ROUTE configuration statement in the OMPROUTE configuration file.

SBNT Indicates that the network is subnetted; such an entry is a placeholder only.

DIR Indicates a directly connected network, subnet, or host.

RIP Indicates a route that was learned through the RIP protocol.

STAT Indicates a nonreplaceable statically configured route.

SPF Indicates that the route is an OSPF intra-area route.

SPIA Indicates that the route is an OSPF interarea route.

SPE1 Indicates OSPF external routes (type 1).

SPE2 Indicates OSPF external routes (type 2).

RNGE Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.

RSTA Indicates a static route that is defined as replaceable.

An asterisk (*) after the route type indicates that the route has a directly connected backup. A percent sign (%) after the route type indicates that RIP updates are always accepted for this destination.

DISTANCE
Indicates the route cost.

Table 4. OMPROUTE IPv4 Route Type and DISTANCE Value mapping

Route Type	Value
SPF or SPIA	The OSPF cost of the route.
SPE1	The OSPF cost to get to the AS boundary router or forwarding address that is used to reach the destination, plus the external cost.

Table 4. OMPROUTE IPv4 Route Type and DISTANCE Value mapping (continued)

Route Type	Value
SPE2	The external cost.
RIP	The RIP metric.
STAT or RSTA	• 0 when the route is direct. • 1 when the route is indirect.
DIR or SBNT	1
RNGE	The OSPF cost of the range.
DFLT	0

AGE Indicates the time that has elapsed since the routing table entry was last refreshed. For routes that have the route type DEL or RIP, this value increments by a factor of 10 for each 10–second increase in age. If the TCP/IP stack enters a storage shortage condition, all routes that have the route type DEL or RIP are refreshed when the shortage condition is relieved.

NEXT HOP(S)
Indicates the IP address of the next router and the interface used to reach that router for each of the paths toward the destination.

All OMPROUTE IPv4 policy-based routing tables:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,PRTABLE=ALL command displays all of the routes in all of the OMPROUTE IPv4 policy-based routing tables. The dynamic routing parameters configured to the Policy Agent for each table are displayed following the routes for that table. A sample output with an explanation of the entries follows.

Results:

- This command displays the contents of the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.
- If a policy-based routing table is configured with no IPv4 dynamic routing parameters, OMPROUTE has no knowledge of that routing table for IPv4. The routing table is not included in the display of OMPROUTE IPv4 policy-based routing tables.

```
EZZ7847I ROUTING TABLE 796
TABLE NAME:     SECLOW1
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   3.0.0.0          FF000000  1       1549    NONE
 SPF   3.3.3.0          FFFFFFFC  2       1561    9.67.102.3
 SPF   3.3.3.3          FFFFFFFF  2       1561    9.67.102.3
 SPF   9.67.101.4       FFFFFFFF  2       1561    9.67.102.3
 DIR*  9.67.102.0       FFFFFF00  1       1575    9.67.102.7
 SPF   9.67.102.3       FFFFFFFF  1       1566    9.67.102.3
 SPF   9.67.102.7       FFFFFFFF  2       1561    CTC7TO3
SPE2   130.200.1.1      FFFFFFFF  0       1379    9.67.102.3
                       0 NETS DELETED
DYNAMIC ROUTING PARAMETERS:
  INTERFACE: CTC7TO3       NEXT HOP: 9.67.102.3

TABLE NAME:     SECLOW2
```

```
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   8.0.0.0          FF000000  1       1549    NONE
 SPF   8.8.8.8          FFFFFFFC  2       1545    9.67.100.8
 SPF   8.8.8.8          FFFFFFFF  2       1545    9.67.100.8
SBNT   9.0.0.0          FF000000  1       1368    NONE
 DIR*  9.67.100.0       FFFFFF00  1       1576    9.67.100.7
 SPF   9.67.100.7       FFFFFFFF  2       1545    CTC7TO8
 SPF   9.67.100.8       FFFFFFFF  1       1572    9.67.100.8
 SPF   9.67.105.4       FFFFFFFF  2       1545    9.67.100.8
SPE2   130.200.0.0      FFFF0000  0       1379    9.67.100.8       (2)
SPE2   130.200.1.18     FFFFFFFF  0       1379    9.67.100.8
SPE2   130.201.0.0      FFFF0000  0       1379    9.67.100.8       (2)
SPE2   130.202.0.0      FFFF0000  0       1379    9.67.100.8       (2)
                       0 NETS DELETED
DYNAMIC ROUTING PARAMETERS:
  INTERFACE:  CTC7TO8       NEXT HOP: 9.67.100.8
  INTERFACE:  CTC7TO8       NEXT HOP: 9.67.100.15
  INTERFACE: *CTC7TO9       NEXT HOP: 9.67.201.53
```

TABLE NAME
Indicates the name of the policy-based routing table.

INTERFACE
Indicates the name of an interface that is specified in a dynamic routing parameter for the policy-based routing table. If the interface is not currently defined to the TCP/IP stack as an IPv4 interface or the interface is inactive on the TCP/IP stack, the name is preceded by an asterisk (*).

NEXT HOP
Indicates the next hop router IP address that is specified in a dynamic routing parameter for the policy-based routing table. The value ANY is displayed when no next-hop router IP address is specified for the dynamic routing parameter.

See “OMPROUTE IPv4 main routing table” on page 57 for additional field descriptions.

OMPROUTE IPv4 policy-based routing table:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,PRTABLE=*prname* command displays all of the routes in a single OMPROUTE IPv4 policy-based routing table. The dynamic routing parameters configured to the Policy Agent for the table are displayed following the routes for the table. A sample output with explanation of entries follows.

Results:

- This command displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.
- If a policy-based route table is configured with no IPv4 dynamic routing parameters, OMPROUTE has no knowledge of that route table for IPv4. You cannot use that route table with this command.

```
EZZ7847I ROUTING TABLE 796
TABLE NAME:     SECLOW2
TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)

SBNT   8.0.0.0          FF000000  1       1549    NONE
 SPF   8.8.8.8          FFFFFFFC  2       1545    9.67.100.8
 SPF   8.8.8.8          FFFFFFFF  2       1545    9.67.100.8
SBNT   9.0.0.0          FF000000  1       1368    NONE
```

```
 DIR*  9.67.100.0       FFFFFF00  1       1576    9.67.100.7
 SPF   9.67.100.7       FFFFFFFF  2       1545    CTC7TO8
 SPF   9.67.100.8       FFFFFFFF  1       1572    9.67.100.8
 SPF   9.67.105.4       FFFFFFFF  2       1545    9.67.100.8
SPE2   130.200.0.0      FFFF0000  0       1379    9.67.100.8       (2)
SPE2   130.200.1.18     FFFFFFFF  0       1379    9.67.100.8
SPE2   130.201.0.0      FFFF0000  0       1379    9.67.100.8       (2)
SPE2   130.202.0.0      FFFF0000  0       1379    9.67.100.8       (2)
                       0 NETS DELETED, 0 NETS INACTIVE
DYNAMIC ROUTING PARAMETERS:
  INTERFACE:  CTC7TO8       NEXT HOP: 9.67.100.8
  INTERFACE:  CTC7TO8       NEXT HOP: 9.67.100.15
  INTERFACE: *CTC7TO9       NEXT HOP: 9.67.201.53
```

See “All OMPROUTE IPv4 policy-based routing tables” on page 61 for field descriptions.

Route expansion information for OMPROUTE IPv4 policy-based routing table: Use the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,PRTABLE=*prname*,DEST=*ip-addr* command to obtain information about a particular route in an OMPROUTE IPv4 policy-based routing table. When multiple equal-cost routes exist, use this command to obtain a list of the next hops. A sample output with explanation of entries follows.

Results:

- This command displays information from the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.
- If a policy-based route table is configured with no IPv4 dynamic routing parameters, OMPROUTE has no knowledge of that route table for IPv4. You cannot use that route table with this command.

```
EZZ7874I ROUTE EXPANSION 370
TABLE NAME:     SECHIGH
DESTINATION:    9.68.101.0
MASK:           255.255.255.0
ROUTE TYPE:     SPF
DISTANCE:       6
AGE:            1344
NEXT HOP(S):    9.167.100.17      (CTC2)
                9.168.100.4       (CTC1)
```

TABLE NAME
    Indicates the name of the policy-based routing table.

See “Route expansion information for the OMPROUTE IPv4 main routing table” on page 59 for additional field descriptions.

Route expansion information for all OMPROUTE IPv4 policy-based routing tables: Use the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,PRTABLE=ALL,DEST=*ip-addr* command to obtain information from all of the OMPROUTE IPv4 policy-based routing tables about a particular route. When multiple equal-cost routes exist in a table, use this command to obtain a list of the next hops. A sample output with explanation of entries follows.

Results:

- This command displays information from the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see "DISPLAY TCPIP,,NETSTAT" on page 9.
- If a policy-based route table is configured with no IPv4 dynamic routing parameters, OMPROUTE has no knowledge of that route table for IPv4. The route table does not appear in the display of OMPROUTE IPv4 route tables.

```
EZZ7874I ROUTE EXPANSION 370
TABLE NAME:     SECHIGH
DESTINATION:    9.68.101.0
MASK:           255.255.255.0
ROUTE TYPE:     SPF
DISTANCE:       6
AGE:            1344
NEXT HOP(S):    9.167.100.17      (CTC2)
                9.168.100.4       (CTC1)

TABLE NAME:     SECLOW
DESTINATION:    9.68.101.0
MASK:           255.255.255.0
ROUTE TYPE:     SPF
DISTANCE:       9
AGE:            2854
NEXT HOP(S):    9.169.102.1       (CTC3)
```

TABLE NAME

Indicates the name of the policy-based routing table.

See "Route expansion information for the OMPROUTE IPv4 main routing table" on page 59 for additional field descriptions.

Deleted OMPROUTE IPv4 routes:

The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,DELETED command displays the routes that have been deleted from the OMPROUTE IPv4 main routing table and that have not been replaced or recycled through garbage collection (garbage collection occurs only when RIP is running). A sample output follows. Explanation of entries is the same as for the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE command (see "OMPROUTE IPv4 main routing table" on page 57).

The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RTTABLE,PRTABLE=*prname*,DELETED command displays the routes that have been deleted from an OMPROUTE IPv4 policy-based routing table and that have not been replaced or recycled through garbage collection.

```
D TCPIP,TCPCS6,OMPROUTE,RTTABLE,DELETED
 EZZxxxxI IPV4 DELETED ROUTES
 TYPE   DEST NET         MASK      COST    AGE     NEXT HOP(S)
  DEL   1.2.3.4          FFFFFFFF  16      12      NONE
    1 NETS DELETED, 1 NETS INACTIVE
```

Examples using the IPV6OSPF command:

All IPv6 OSPF information:

The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,ALL command displays a comprehensive list of IPv6 OSPF information. A sample output with explanation of entries follows:

```
EZZ7970I IPV6 OSPF INFORMATION 322
TRACE6: 0, DEBUG6: 0
STACK AFFINITY            TCPCS67
IPV6 OSPF PROTOCOL:       ENABLED
IPV6 OSPF ROUTER ID:      67.67.67.67  (*IPV6_OSPF)
DFLT IPV6 OSPF INST ID:   0
EXTERNAL COMPARISON:      TYPE 2
AS BOUNDARY CAPABILITY:   ENABLED
IMPORT EXTERNAL ROUTES:   RIP
ORIG. DEFAULT ROUTE:      NO
DEMAND CIRCUITS:          ENABLED
DR MAX ADJ ATTEMPT:       10

EZZ7973I IPV6 OSPF AREAS
AREA ID         STUB DFLT-COST IMPORT-PREF DEMAND IFCS NETS RTRS ABRS
6.6.6.6          NO        N/A     N/A       OFF     2    1    4    2
0.0.0.0          NO        N/A     N/A       OFF     2    0    4    2

--AREA RANGES--
AREA ID         ADVERTISE  PREFIX
6.6.6.6             NO     2001:DB8:0:101::/64

EZZ7958I IPV6 OSPF INTERFACES
NAME             AREA            TYPE   STATE COST HELLO DEAD NBRS ADJS
VIPA1A6          6.6.6.6         VIPA    N/A     1  N/A   N/A  N/A  N/A
MPCPTP7TO5       0.0.0.0         P-2-MP   16     1   10    40    1    1
NSQDIO1L6        6.6.6.6         BRDCST   32     1   10    40    3    2
VL/0             0.0.0.0         VLINK    16     1   30   180    1    1

EZZ7972I IPV6 OSPF VIRTUAL LINKS
ENDPOINT        TRANSIT AREA   STATE COST HELLO DEAD NBRS ADJS
64.64.64.64     6.6.6.6          16     1   30   180    1    1

EZZ8129I IPV6 OSPF NEIGHBORS
ROUTER ID       STATE LSRXL DBSUM LSREQ HSUP RTR-PRI IFC
65.65.65.65       128     0     0     0  OFF       1 MPCPTP7TO5
64.64.64.64       128     0     0     0  OFF       1 NSQDIO1L6
63.63.63.63       128     0     0     0  OFF       1 NSQDIO1L6
68.68.68.68       128     0     0     0  OFF       1 NSQDIO1L6
64.64.64.64       128     0     0     0  OFF       1 *
```

TRACE6
Displays the level of tracing currently in use by OMPROUTE IPv6 routing protocols.

DEBUG6
Displays the level of debugging currently in use by OMPROUTE IPv6 routing protocols.

STACK AFFINITY
Displays the name of the stack on which OMPROUTE is running.

IPV6 OSPF PROTOCOL
Displays whether IPv6 OSPF is enabled or disabled.

IPV6 OSPF ROUTER ID
Displays the IPv6 OSPF Router ID and its configuration source. Possible sources are:

- OMPROUTE configuration statement (denoted by a prefixed asterisk "*") that has the RouterID parameter specified:
  1. IPV6_OSPF
  2. ROUTERID (if the IPv6 router ID was inherited from the router ID specified for IPv4)

3. OSPF (if the IPv6 router ID was inherited from the router ID specified for IPv4)

- The name of the IPv4 interface that was used by OMPROUTE to set the router ID. This indicates that you did not specify an IPv6 router ID, so the IPv6 router ID was inherited from the IPv4 router ID, which had been defaulted by OMPROUTE to the IP address assigned to an IPv4 interface.

For more information about assigned and configured router IDs, see Steps for configuring OSPF and RIP (IPv4 and IPv6) in the z/OS Communications Server: IP Configuration Guide.

DFLT IPV6 OSPF INST ID
Displays the default value for the OSPF protocol instance identifier for IPV6_OSPF_INTERFACEs.

EXTERNAL COMPARISON
Displays the external route type used by IPv6 OSPF when importing external information into the IPv6 OSPF domain and when comparing IPv6 OSPF external routes to IPv6 RIP routes.

AS BOUNDARY CAPABILITY
Indicates whether external routes are imported into the IPv6 OSPF domain.

IMPORT EXTERNAL ROUTES
Indicates the types of external routes that are imported into the IPv6 OSPF domain. Displayed only when AS Boundary Capability is enabled.

ORIG DEFAULT ROUTE
Indicates whether a default route is originated into the IPv6 OSPF domain. Orig Default Route is displayed only when AS Boundary Capability is enabled.

DEFAULT ROUTE COST
Displays the cost and type of the default route (if originated). Default Route Cost is displayed only when AS Boundary Capability is enabled and Orig Default Route is Always.

DEFAULT FORWARD ADDR
Displays the forwarding address specified in the default route (if originated). Default Forwarding Address is displayed only when AS Boundary Capability is enabled and Orig Default Route is Always.

LEARN HIGHER COST DFLT
Indicates whether IPv6 OSPF will learn default routes from inbound packets when their cost is higher than the default route originated by this host. This parameter is displayed only when AS Boundary Capability is enabled and Orig Default Route is Always.

DEMAND CIRCUITS
Indicates whether demand circuit support is available for IPv6 OSPF interfaces.

DR MAX ADJ ATTEMPT
Establishes a threshold value for maximum number of adjacency attempts to a neighboring designated router. It is used for reporting and controlling futile neighbor state loops. For information about futile neighbor state loops, see the futile neighbor state loops information in the z/OS Communications Server: IP Configuration Guide.

The remainder of the DISPLAY `TCPIP,`*`tcpipjobname`*`,OMPROUTE,IPV6OSPF,ALL` output is described in the following sections.

IPv6 OSPF area statistics and parameters:

The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,AREASUM command displays the statistics and parameters for all IPv6 OSPF areas attached to the router. A sample output with an explanation of entries follows:

```
EZZ7973I IPV6 OSPF AREAS 536
AREA ID         STUB DFLT-COST IMPORT-PREF DEMAND IFCS NETS RTRS ABRS
6.6.6.6          NO        N/A     N/A       OFF     2    1    4    2
0.0.0.0          NO        N/A     N/A       OFF     2    0    4    2

--AREA RANGES--
AREA ID         ADVERTISE  PREFIX
6.6.6.6             NO     2001:DB8:0:101::/64
```

AREA ID
Indicates the ID of the area.

STUB
Indicates whether the area is a stub area.

DFLT-COST
Displays the cost of the default route configured for the stub area.

IMPORT-PREF
Indicates whether Inter-Area Prefix LSAs are to be imported into the stub area.

DEMAND
Indicates whether demand circuits are supported in this area. This is ON when every router in the area supports demand circuits, otherwise it is OFF.

IFCS
Indicates the number of router interfaces attached to the particular area. These interfaces are not necessarily functional.

NETS
Indicates the number of transit networks that have been found while doing the SPF tree calculation for this area.

RTRS
Indicates the number of routers that have been found when doing the SPF tree calculation for this area.

ABRS
Indicates the number of area border routers that have been found when doing the SPF tree calculation for this area.

AREA RANGES
Indicates that information about ranges configured for this area follows.

ADVERTISE
Indicates whether a given range within an area is to be advertised into other areas.

PREFIX
Displays the prefix and prefix length for a given range within an area.

IPv6 OSPF interface statistics and parameters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,INTERFACE,NAME=*if-name*,ID=*if-id* command displays current, run-time statistics and parameters related to IPv6 OSPF interfaces. (The keyword IF can be substituted for INTERFACE.) Either the NAME= parameter or the ID= parameter can be specified, but not both. If no NAME= or ID= parameter is given (see Example 1), a single line is printed summarizing each interface. If NAME= or ID= parameter is given (see Example 2), detailed statistics for that interface are displayed. Sample outputs with an explanation of entries follow:

```
----Example 1 ----
EZZ7958I IPV6 OSPF INTERFACES 575
NAME             AREA            TYPE   STATE COST HELLO DEAD NBRS ADJS
VIPA1A6          6.6.6.6         VIPA    N/A     1  N/A   N/A  N/A  N/A
MPCPTP7TO5       0.0.0.0         P-2-MP   16     1   10    40    1    1
NSQDIO1L6        6.6.6.6         BRDCST   32     1   10    40    3    2
OSAGBE1          3.3.3.3         BRDCST   32     1   10    40    4    2
OSAGBE2          3.3.3.3         BRDCST    2     1   10    40    0    0
VL/0             0.0.0.0         VLINK    16     1   30   180    1    1
```

NAME

Displays the interface name.

AREA

Attached area ID.

TYPE

Can be one of the following value:

TYPE	Description
BRDCST	Broadcast interface
P-2-MP	Point-to-multipoint interface
VLINK	OSPF virtual link
VIPA	Virtual IP address link

STATE

Can be one of the following vaule:

STATE	Description
1	Down
1*	Suspend - This state is not described in RFC2328. The interface is suspended because of a MODIFY command or because it was unable to establish an adjacency with a neighboring designated router after having exceeded the futile neighbor state loop threshold (DR_Max_Adj_Attempt). For information on futile neighbor state loops, see the futile neighbor state loops information in the z/OS Communications Server: IP Configuration Guide.
2	Backup
4	Looped back
8	Waiting
16	Point-to-point
32	DR other
64	Backup DR
128	Designated router

For more information about these values, see RFC 1583 (OSPF Version 2).

COST

Indicates the cost (or metric) associated with the interface.

HELLO

Indicates the number of seconds between Hello packets sent from the interface.

DEAD

Indicates the number of seconds after not having received an OSPF Hello packet, that a neighbor is declared to be down.

NBRS
Number of neighbors. This is the number of routers whose hellos have been received.

ADJS
Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

```
----Example 2 ----
EZZ7959I IPV6 OSPF INTERFACE DETAIL 677
INTERFACE NAME:     NSQDIO1L6
INTERFACE ID:       20
INSTANCE ID:        0
INTERFACE ADDRESS:  FE80::7
                    2001:DB8:0:120::7
INTERFACE PREFIX:   STAT 2001:DB8:0:120::/64
ATTACHED AREA:      6.6.6.6
INTERFACE TYPE:     BRDCST
STATE:              32
DESIGNATED ROUTER:  68.68.68.68
BACKUP DR:          64.64.64.64

DR PRIORITY:       1  HELLO INTERVAL:   10  RXMT INTERVAL:     5
DEAD INTERVAL:    40  TX DELAY:          1  POLL INTERVAL:   N/A
DEMAND CIRCUIT:  OFF  HELLO SUPPRESS:  N/A  SUPPRESS REQ:    N/A
MTU:            9000  COST:              1  DB_EX INTERVAL:   40

# NEIGHBORS:       3  # ADJACENCIES:     2  # FULL ADJS.:      2
# MCAST FLOODS:    7  # MCAST ACKS:      9  # MAX ADJ. RESETS: 0
# ERR PKTS RCVD:   0

NETWORK CAPABILITIES:
 BROADCAST
 DEMAND-CIRCUITS
 MULTICAST
```

INTERFACE NAME
Displays the interface name.

INTERFACE ID
Number that uniquely identifies the interface among the collection of all OSPF interfaces on this TCP/IP stack.

INSTANCE ID
The IPv6 OSPF Instance ID for this interface.

INTERFACE ADDRESS
Indicates the IP addresses that have been learned from the TCP/IP stack for the interface.

INTERFACE PREFIX
Lists the prefixes of the interface. RADV indicates the prefix was learned through IPv6 Router Discovery. STAT indicates it was statically defined to this interface using the PREFIX parameter of the IPV6_OSPF_INTERFACE statement. OSPF indicates it was learned using the OSPF protocol.

ATTACHED AREA
Attached area ID.

INTERFACE TYPE
Can be one of the following value:

INTERFACE TYPE	Description
BRDCST	Broadcast interface

INTERFACE TYPE	Description
P-2-MP	Point-to-multipoint interface
VLINK	OSPF virtual link
VIPA	Virtual IP address link

STATE
Can be one of the following value:

STATE	Description
1	Down
1*	Suspend - This state is not described in RFC2328. The interface is suspended because of a MODIFY command or because it was unable to establish an adjacency with a neighboring designated router after having exceeded the futile neighbor state loop threshold (DR_Max_Adj_Attempt). For information on futile neighbor state loops, see the futile neighbor state loops information in the z/OS Communications Server: IP Configuration Guide.
2	Backup
4	Looped back
8	Waiting
16	Point-to-point
32	DR other
64	Backup DR
128	Designated router

For more information about these values, see RFC 1583 (OSPF Version 2).

DESIGNATED ROUTER
Router ID of the designated router.

BACKUP DR
Router ID of the backup designated router.

DR PRIORITY
Displays the interface router priority used when selecting the designated router. A higher value indicates that this OMPROUTE is more likely to become the designated router. A value of 0 indicates that OMPROUTE never becomes the designated router.

HELLO INTERVAL
Indicates the number of seconds between Hello packets sent from the interface.

RXMT INTERVAL
Displays the frequency (in seconds) of retransmitting link state update packets, link state request packets, and database description packets.

DEAD INTERVAL
Indicates the number of seconds after not having received an OSPF Hello packet, that a neighbor is declared to be down.

TX DELAY
Displays the transmission delay value (in seconds). As each link state advertisement is sent out through this interface, it is aged by this value.

POLL INTERVAL
Displays the poll interval value.

DEMAND CIRCUIT
Displays the current demand circuit status.

HELLO SUPPRESS
Displays whether Hello Suppression is currently on or off.

SUPPRESS REQ
Displays whether Hello Suppression was requested for this interface.

MTU
Indicates the value of the Maximum Transmission Unit.

COST
Indicates the cost (or metric) associated with the interface.

DB_EX INTERVAL
Indicates the number of seconds to allow the database exchange to complete.

# NEIGHBORS
Number of neighbors. This is the number of routers whose hellos have been received.

# ADJACENCIES
Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

# FULL ADJS
Number of full adjacencies. This is the number of neighbors whose state is Full (and therefore with which the router has synchronized databases).

# MAX ADJ. RESETS
The total number of times that the maximum threshold value for adjacency attempts (see the DR MAX ADJ ATTEMPT field) with a neighboring designated router has been reset. A value of N/A indicates that the field is not applicable for that interface, based on the interface type that is used to reach a neighbor. See the types of interfaces supported by OMPROUTE information in z/OS Communications Server: IP Configuration Reference for the types of interfaces that support the futile neighbor state loop detection for OSPF.

# MCAST FLOODS
Number of link state updates that flooded the interface (not counting retransmissions).

# MCAST ACKS
Number of link state acknowledgments that flooded the interface (not counting retransmissions).

# ERR PKTS RCVD
Number of the packets received on the interface that contain errors. These errors include bad packet type, bad length, bad checksum, or other errors.

NETWORK CAPABILITIES
Displays the capabilities of the interface.

IPv6 OSPF virtual link statistics and parameters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,VLINK,ENDPT=*router-id* command displays current, run-time statistics and parameters related to IPv6 OSPF virtual links. If no ENDPT= parameter is given (see Example 1), a single line is printed summarizing each virtual link. If ENDPT= parameter is given (see Example 2), detailed statistics for that virtual link is displayed. Sample outputs with an explanation of entries follow:

```
----Example 1 ----
EZZ7972I IPV6 OSPF VIRTUAL LINKS 703
ENDPOINT        TRANSIT AREA   STATE COST HELLO DEAD NBRS ADJS
64.64.64.64     6.6.6.6          16     1   30   180    1    1
```

ENDPOINT
Indicates the router ID of the virtual neighbor (other endpoint).

TRANSIT AREA
Indicates the non-backbone, non-stub area through which the virtual link is configured.

STATE
Can be one of the following value:

STATE	Description
1	Down
16	Point-to-point

For more information about these values, see RFC 1583 (OSPF Version 2).

COST
Indicates the cost (or metric) associated with the virtual link.

HELLO
Indicates the number of seconds between Hello packets sent from the virtual link.

DEAD
Indicates the number of seconds after not having received an OSPF Hello packet, that a neighbor is declared to be down.

NBRS
Number of neighbors. This is the number of routers whose hellos have been received.

ADJS
Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

```
----Example 2 ----
EZZ7971I IPV6 VIRTUAL LINK DETAILS 713
VIRTUAL LINK ENDPOINT:     64.64.64.64
PHYSICAL INTERFACE NAME:   NSQDIO1L6
VL TRANSIT AREA:           6.6.6.6
STATE:                     16

HELLO INTERVAL:     30  DEAD INTERVAL:     180  DB_EX INTERVAL:    180
RXMT INTERVAL:      10  TX DELAY:            5  COST:                1
DEMAND CIRCUIT:     ON  HELLO SUPPRESS:    OFF  SUPPRESS REQ:       ON

# NEIGHBORS:         1  # ADJACENCIES:       1  # FULL ADJS.:        1
```

VIRTUAL LINK ENDPOINT
Indicates the router ID of the virtual neighbor (other endpoint).

PHYSICAL INTERFACE NAME
Indicates the name of the physical interface being used by the virtual link.

VL TRANSIT AREA
Indicates the non-backbone, non-stub area through which the virtual link is configured.

STATE
Can be one of the following value:

STATE	Description
1	Down
16	Point-to-point

For more information about these values, see RFC 1583 (OSPF Version 2).

HELLO INTERVAL
Indicates the number of seconds between Hello packets sent from the virtual link.

DEAD INTERVAL
Indicates the number of seconds after not having received an OSPF Hello packet, that a neighbor is declared to be down.

DB_EX INTERVAL
Indicates the number of seconds to allow the database exchange to complete.

RXMT INTERVAL
Displays the frequency (in seconds) of retransmitting link state update packets, link state request packets, and database description packets.

TX DELAY
Displays the transmission delay value (in seconds). As each link state advertisement is sent out through this interface, it is aged by this value.

COST
Indicates the cost (or metric) associated with the virtual link.

DEMAND CIRCUIT
Displays the current demand circuit status.

HELLO SUPPRESS
Displays whether Hello Suppression is currently on or off.

SUPPRESS REQ
Displays whether Hello Suppression was requested for this interface.

# NEIGHBORS
Number of neighbors. This is the number of routers whose hellos have been received.

# ADJACENCIES
Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

# FULL ADJS
Number of full adjacencies. This is the number of neighbors whose state is Full (and therefore with which the router has synchronized databases).

IPv6 OSPF neighbor statistics and parameters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,NEIGHBOR,ID=*router-id*,IFNAME=*if_name* command displays the statistics and parameters related to IPv6 OSPF neighbors. (The keyword NBR can be substituted for NEIGHBOR.)

- If no ID= parameter is given (see Example 1), a single line is printed summarizing each neighbor.
- If an ID= parameter is given (see Example 2), detailed statistics for that neighbor are displayed.

- If the neighbor specified by the ID= parameter has more than one neighbor relationship with OMPROUTE (for example if there are multiple IPv6 OSPF links connecting them), the IFNAME= parameter can be used to specify which link's adjacency to examine (for an adjacency over a virtual link, specify IFNAME=*).

See the following sample outputs with an explanation of entries:

```
----Example 1 ----
EZZ8129I IPV6 OSPF NEIGHBORS 715
ROUTER ID       STATE LSRXL DBSUM LSREQ HSUP RTR-PRI IFC
65.65.65.65       128     0     0     0  OFF       1 MPCPTP7TO5
63.63.63.63         8     0     0     0  OFF       1 NSQDIO1L6
64.64.64.64       128     0     0     0  OFF       1 NSQDIO1L6
68.68.68.68       128     0     0     0  OFF       1 NSQDIO1L6
64.64.64.64       128     0     0     0  OFF       1 *
```

ROUTER ID
Displays the neighbor's OSPF router ID.

STATE
Can be one of the following value:

STATE	Description
1	Down
2	Attempt
4	Init
8	2–Way
16	ExStart
32	Exchange
64	Loading
128	Full

For more information about these values, see RFC 1583 (OSPF Version 2).

LSRXL
Displays the size of the current link state retransmission list for this neighbor.

DBSUM
Displays the size of the database summary list waiting to be sent to the neighbor.

LSREQ
Displays the number of link state advertisements that are being requested from the neighbor.

HSUP
Displays whether hello suppression is active with the neighbor.

RTR-PRI
Displays the neighbor's router priority. Higher router priority indicates that it is more likely to become a designated router. A router priority of 0 indicates that the neighbor is not eligible to become designated router. N/A indicates the neighbor is not on a multi-access link; therefore, no designated router is required.

IFC
Displays the name of the interface over which a relationship has been established with this neighbor. An asterisk (*) displayed in this column indicates that the neighbor relationship has been established over a virtual link.

```
----Example 2 ----
EZZ8130I IPV6 OSPF NEIGHBOR DETAILS 737
NEIGHBOR IP ADDRESS:    FE80::4
OSPF ROUTER ID:         64.64.64.64
NEIGHBOR STATE:         128
PHYSICAL INTERFACE:     NSQDIO1L6
DR CHOICE:              68.68.68.68
BACKUP CHOICE:          64.64.64.64
DR PRIORITY:            1
NBR OPTIONS:            V6,E,R (0X0013)

DB SUMM QLEN:      0  LS RXMT QLEN:      0  LS REQ QLEN:       0
LAST HELLO:        5  NO HELLO:        OFF
# LS RXMITS:       1  # DIRECT ACKS:     5  # DUP LS RCVD:     4
# OLD LS RCVD:     0  # DUP ACKS RCVD:   3  # ADJ. RESETS:     1
# ERR LS RCVD:     0
```

NEIGHBOR IP ADDRESS
Displays the link-local IP address of the neighbor's interface to the common link.

OSPF ROUTER ID
Displays the neighbor's OSPF router ID.

NEIGHBOR STATE
Can be one of the following value:

NEIGHBOR STATE	Description
1	Down
2	Attempt
4	Init
8	2–Way
16	ExStart
32	Exchange
64	Loading
128	Full

For more information about these values, see RFC 1583 (OSPF Version 2).

PHYSICAL INTERFACE
Displays the name of the interface over which a relationship has been established with this neighbor.

DR CHOICE, BACKUP CHOICE, DR PRIORITY
Indicate the values seen in the last hello message received from the neighbor. N/A indicates that the neighbor is not on a multiaccess link; therefore, no designated router is required.

NBR OPTIONS
Indicates the optional OSPF capabilities supported by the neighbor. These capabilities are denoted by:

NBR OPTIONS	Description
V6	The router can be used in IPv6 routing calculations.
E	Processes AS External LSAs. When this is not set, the area to which the common network belongs has been configured as a stub.
MC	RFC 1584 (Multicast Extensions to OSPF) is supported. This value is never set by OMPROUTE but can be received from other routers.

NBR OPTIONS	Description
N	Describes the handling of Type-7 LSAs - Multicast OSPF. This value is never set by OMPROUTE but might be received from other routers.
R	Is an active router. Routes that transit the neighbor can be computed.
DC	RFC 1793 (Extending OSPF to Support Demand Circuits) is supported.

This field is valid only for those neighbors in state Exchange or greater.

DB SUMM QLEN
Indicates the number of advertisements waiting to be summarized in Database Description packets. It must be 0 except when the neighbor is in state Exchange.

LS RXMT QLEN
Indicates the number of advertisements that have been flooded to the neighbor, but not yet acknowledged.

LS REQ QLEN
Indicates the number of advertisements that are being requested from the neighbor in state Loading.

LAST HELLO
Indicates the number of seconds since a hello message has been received from the neighbor. If the TCP/IP stack enters a storage shortage condition, this value is reset to 0 when the shortage condition is relieved.

NO HELLO
Indicates whether Hello Suppression is active with the neighbor.

# LS RXMITS
Indicates the number of retransmissions that have occurred during flooding.

# DIRECT ACKS
Indicates the number of acknowledgements sent in response to duplicate link state advertisements.

# DUP LS RCVD
Indicates the number of duplicate retransmissions that have occurred during flooding.

# OLD LS RCVD
Indicates the number of old advertisements received during flooding.

# DUP ACKS RCVD
Indicates the number of duplicate acknowledgments received.

# ADJ. RESETS
Indicates the number of times the neighbor has transitioned down to ExStart state.

# ERR LS RVCD
Number of the link state advertisements received from the neighbor that are unexpected or that contain errors. These errors include bad advertisement type, bad length, bad checksum, or other errors.

IPv6 OSPF link state database statistics:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,DBSIZE command displays the number of LSAs currently in the link state database, categorized by type. The following example is a sample output:

```
EZZ8128I IPV6 OSPF LS DATABASE SIZE 841
# ROUTER-LSAS:             8
# NETWORK-LSAS:            1
# INTER-AREA PREFIX LSAS:  50
# INTER-AREA ROUTER LSAS:  6
# AS EXTERNAL-LSAS:        6
# LINK LSAS:               6
# INTRA-AREA PREFIX LSAS:  21
# UNKNOWN LSAS:            0
# INTRA-AREA ROUTES:       24
# INTER-AREA ROUTES:       0
# TYPE 1 EXTERNAL ROUTES:  0
# TYPE 2 EXTERNAL ROUTES:  0
```

IPv6 OSPF link state advertisement:
The following command displays the contents of a single link state advertisement contained in the IPv6 OSPF database:

DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,LSA,LSTYPE=*ls-type*,LSID=*lsid*,ORIG=*ad-router*,AREAID=*area-id*,IFNAME=*if_name*

For a summary of all non-external advertisements in the IPv6 OSPF database, use the following command: DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,DATABASE,AREAID=*area-id*

For a summary of all external advertisements in the IPv6 OSPF database, use the following command: DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,EXTERNAL

The following example shows a sample output of a Router LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 834
        LS AGE:          61
        LS TYPE:         0X2001 (ROUTER LSA)
        LS ID:           0
        LS ORIGINATOR:   64.64.64.64
        LS SEQUENCE NO:  0X8000000F
        LS CHECKSUM:     0X3886
        LS LENGTH:       40
        ROUTER TYPE:     (0X01) ABR
        LS OPTIONS:      (0X000033) V6,E,R,DC
INTERFACES:
 TYPE  METRIC  INTERFACE ID   NBR INTERFACE ID   NBR ROUTER ID
    2       1            16                 14     68.68.68.68
```

LS AGE
The time, in seconds, since the LSA was originated. An asterisk (*) displayed beside the age value indicates that the originator is supporting demand circuits and has indicated that this LSA should not be aged.

LS TYPE
Classifies the advertisement and dictates its contents. LS Type values are hexadecimal values.

LS TYPE	Description
0x2001	Router LSA, has area scope.
0x2002	Network LSA, has area scope.
0x2003	Inter-Area Prefix LSA, has area scope.
0x2004	Inter-Area Router LSA, has area scope.

LS TYPE	Description
0x4005	AS External LSA, has global scope throughout the IPv6 OSPF autonomous sytem.
0x0008	Link LSA, has link scope.
0x2009	Intra-Area Prefix LSA, has area scope.

LS ID
Together with LS Type and LS Originator, uniquely identifies the LSA in the link state database.

LS ORIGINATOR
The Router ID of the router that originated the LSA.

LS SEQUENCE NO
Used to detect old or duplicate LSAs. Successive instances of an LSA are given successive LS sequence numbers.

LS CHECKSUM
The Fletcher checksum of the complete contents of the LSA, including the LSA header but excluding the LS age field.

LS LENGTH
The length in bytes of the LSA, including the 20–byte LSA header.

ROUTER TYPE
Indicates the level of function of the advertising router and can be one of the following type:

ROUTER TYPE	Description
ASBR	The router is an AS boundary router.
ABR	The router is an area border router.
V	The router is an endpoint of one of more fully adjacent virtual links having the described area as transit area.
W	The router is a wildcard multicast receiver (OMPROUTE will never set the W option on its own Router LSAs).

LS OPTIONS
Indicates the optional OSPF capabilities supported by the piece of the routing domain described by the advertisement, denoted by:

LS OPTIONS	Description
V6	The information in the LSA can be used in IPv6 routing calculations.
E	Processes AS External LSAs. When this is not set, the area to which the advertisement belongs has been configured as a stub.
MC	RFC 1584 (Multicast Extensions to OSPF) is supported. This value is never set by OMPROUTE but can be received from other routers.
N	Describes the handling of Type-7 LSAs - Multicast OSPF. This value is never set by OMPROUTE but can be received from other routers.
R	Routes can be computed which transit the advertising node.
DC	RFC 1793 (Extending OSPF to Support Demand Circuits) is supported.

INTERFACES
Subheader indicating that information about interfaces advertised on this Router LSA follows.

TYPE
The kind of interface being described:

TYPE	Description
1	Point-to-point connection to another router
2	Connection to a transit network
4	Virtual link

METRIC
The cost of using this router interface, for outbound traffic.

INTERFACE ID
The interface ID assigned to the interface being described.

NBR INTERFACE ID
The interface ID that the neighbor router (or, for Type 2 interfaces, the link's designated router) has been advertising in hello packets sent on the link.

NBR ROUTER ID
The Router ID of the neighbor router, or, for Type 2 interfaces, the link's designated router.

The following example shows a sample output of a Network LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 877
        LS AGE:          268
        LS TYPE:         0X2002 (NETWORK LSA)
        LS ID:           14
        LS ORIGINATOR:   68.68.68.68
        LS SEQUENCE NO:  0X80000003
        LS CHECKSUM:     0X774C
        LS LENGTH:       40
        LS OPTIONS:      (0X000033) V6,E,R,DC
ATTACHED ROUTERS:
 68.68.68.68      67.67.67.67      64.64.64.64      63.63.63.63
```

LS AGE, LS TYPE, LS ID, LS ORIGINATOR, LS SEQUENCE NO, LS CHECKSUM, LS LENGTH, LS OPTIONS
See descriptions for these values in the Router LSA sample in “IPv6 OSPF link state advertisement” on page 77.

ATTACHED ROUTERS
The Router IDs of each of the routers attached to the link. This includes the Designated Router and all routers that are fully adjacent to the Designated Router.

The following example shows a sample output of an Inter-Area Prefix LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 881
        LS AGE:          58
        LS TYPE:         0X2003 (INTER-AREA PREFIX LSA)
        LS ID:           23
        LS ORIGINATOR:   64.64.64.64
        LS SEQUENCE NO:  0X80000002
        LS CHECKSUM:     0X1C69
        LS LENGTH:       44
        PREFIX:          2001:DB8:0:120::7/128
        PREFIX-OPTIONS:  (0X00)
        METRIC:          1
```

LS AGE, LS TYPE, LS ID, LS ORIGINATOR, LS SEQUENCE NO, LS CHECKSUM, LS LENGTH

See descriptions for these values in the Router LSA sample in “IPv6 OSPF link state advertisement” on page 77.

PREFIX

The prefix being described by the LSA.

PREFIX OPTIONS

The optional capabilities of the prefix including the following values:

PREFIX OPTIONS	Description
NU	The prefix should be excluded from IPv6 unicast calculations.
LA	The prefix is actually an IPv6 interface address of the advertising router.
MC	The prefix should be included in IPv6 multicast routing calculations.
P	On NSSA area prefixes, the prefix should be readvertised at the NSSA area border. OMPROUTE cannot be an NSSA area router.

METRIC

The cost of the route from the LSA originator to the prefix being described by the LSA.

The following example shows a sample output of an Inter-Area Router LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 933
        LS AGE:          *8
        LS TYPE:         0X2004 (INTER-AREA ROUTER LSA)
        LS ID:           2
        LS ORIGINATOR:   64.64.64.64
        LS SEQUENCE NO:  0X80000001
        LS CHECKSUM:     0X9859
        LS LENGTH:       32
        LS OPTIONS:      (0X000033) V6,E,R,DC
        ROUTER ID:       68.68.68.68
        METRIC:          1
```

LS AGE, LS TYPE, LS ID, LS ORIGINATOR, LS SEQUENCE NO, LS CHECKSUM, LS LENGTH, LS OPTIONS

See descriptions for these values in the Router LSA sample in “IPv6 OSPF link state advertisement” on page 77.

ROUTER ID

The Router ID of the router being described by the LSA.

METRIC

The cost of the route from the LSA originator to the router being described by the LSA.

The following example shows a sample output of an AS External LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 207
        LS AGE:          33
        LS TYPE:         0X4005 (AS EXTERNAL LSA)
        LS ID:           4
        LS ORIGINATOR:   67.67.67.67
        LS SEQUENCE NO:  0X80000001
        LS CHECKSUM:     0X4D64
        LS LENGTH:       36
        METRIC:          2
        METRIC TYPE:     2
        PREFIX-OPTIONS:  (0X00)
        PREFIX:          2001:DB8:0:A1B::/64
```

LS AGE, LS TYPE, LS ID, LS ORIGINATOR, LS SEQUENCE NO, LS CHECKSUM, LS LENGTH
See descriptions for these values in the Router LSA sample in "IPv6 OSPF link state advertisement" on page 77.

METRIC
The cost of the route from the LSA originator to the prefix being described by the LSA.

METRIC TYPE
Whether the specified metric is a Type 1 or Type 2 external metric.

PREFIX OPTIONS
The optional capabilities of the prefix including the following values:

PREFIX OPTIONS	Description
NU	The prefix should be excluded from IPv6 unicast calculations.
LA	The prefix is actually an IPv6 interface address of the advertising router.
MC	The prefix should be included in IPv6 multicast routing calculations.
P	On NSSA area prefixes, the prefix should be readvertised at the NSSA area border. OMPROUTE cannot be an NSSA area router.

PREFIX
The prefix being described by the LSA.

FORWARD ADDR
Optional field. If included, data traffic for the advertised destination should be forwarded to this address.

ROUTE TAG
Optional field. If included, communicates additional information between AS boundary routers.

REF TYPE,REF LS ID
Optional fields. If included, additional information concerning the advertised external route can be found in the LSA having LS type of REF TYPE, Link State ID of REF LS ID, and LS Originator the same as specified in this LSA.

The following example shows a sample output of a Link LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 911
        LS AGE:          2
        LS TYPE:         0X0008 (LINK LSA)
        LS ID:           34
        LS ORIGINATOR:   63.63.63.63
        LS SEQUENCE NO:  0X80000003
        LS CHECKSUM:     0X34E8
        LS LENGTH:       56
        LS OPTIONS:      (0X000033) V6,E,R,DC
        LINK LOCAL ADDR: FE80::3
        ROUTER PRIORITY: 1
        # PREFIXES:      1

PREFIX-OPTIONS        PREFIX
(0X00)                2001:DB8:0:120::/64
```

LS AGE, LS TYPE, LS ID, LS ORIGINATOR, LS SEQUENCE NO, LS CHECKSUM, LS LENGTH, LS OPTIONS
See descriptions for these values in the Router LSA sample in “IPv6 OSPF link state advertisement” on page 77.

LINK LOCAL ADDR
The originating router's link-local address on the link.

ROUTER PRIORITY
The router priority of the interface attaching the originating router to the link. Used in electing Designated Router.

# PREFIXES
The number of IPv6 address prefixes contained in the LSA.

PREFIX OPTIONS
The optional capabilities of the prefix:

PREFIX OPTIONS	Description
NU	The prefix should be excluded from IPv6 unicast calculations.
LA	The prefix is actually an IPv6 interface address of the advertising router.
MC	The prefix should be included in IPv6 multicast routing calculations.
P	On NSSA area prefixes, the prefix should be readvertised at the NSSA area border. OMPROUTE cannot be an NSSA area router.

PREFIX
An IPv6 prefix to be associated with the link.

The following example shows a a sample output of an Intra-Area Prefix LSA with an explanation of entries:

```
EZZ7880I LSA DETAILS 913
        LS AGE:          32
        LS TYPE:         0X2009 (INTRA-AREA PREFIX LSA)
        LS ID:           14
        LS ORIGINATOR:   68.68.68.68
        LS SEQUENCE NO:  0X80000004
        LS CHECKSUM:     0X6ECA
        LS LENGTH:       52
        # PREFIXES:      1
        REF LS TYPE:     0X2001
        REF LS ID:       0
        REF ORIG:        68.68.68.68

METRIC  PREFIX-OPTIONS        PREFIX
0       (0X02) LA             2001:DB8:0:120::8/128
```

LS AGE, LS TYPE, LS ID, LS ORIGINATOR, LS SEQUENCE NO, LS CHECKSUM, LS LENGTH
See descriptions for these values in the Router LSA sample in “IPv6 OSPF link state advertisement” on page 77.

# PREFIXES
The number of IPv6 address prefixes contained in the LSA.

REF LS TYPE,REF LS ID,REF ORIG
Identifies the Router LSA or Network LSA with which the IPv6 address prefixes should be associated.

METRIC
The cost of the route from the LSA originator to each of prefixes being described.

PREFIX OPTIONS
The optional capabilities of each of the prefixes being described:

PREFIX OPTIONS	Description
NU	The prefix should be excluded from IPv6 unicast calculations.
LA	The prefix is actually an IPv6 interface address of the advertising router.
MC	The prefix should be included in IPv6 multicast routing calculations.
P	On NSSA area prefixes, the prefix should be readvertised at the NSSA area border. OMPROUTE cannot be an NSSA area router.

PREFIX
The list of prefixes being described.

IPv6 OSPF external advertisements:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,EXTERNAL command lists the AS external advertisements belonging to the IPv6 OSPF routing domain. One line is printed for each advertisement. Each advertisement is defined by the following three parameters:

- Its link state type (always 4005 for AS external advertisements)
- Its link state ID
- The advertising router (called the LS originator)

A sample output with an explanation of entries follows:

```
EZZ8127I IPV6 OSPF AS EXTERNAL LSDB 555
              AS EXTERNAL LSAS (LS TYPE=4005)
LS ORIGINATOR   LS ID     SEQNO       AGE PREFIX
67.67.67.67     5         0X80000001  565 6:6:6:6:6:6:6:6/128
67.67.67.67     6         0X80000001  561 2001:DB8:0:A1C::6/128
67.67.67.67     7         0X80000001  558 2001:DB8:0:103::6/128
67.67.67.67     8         0X80000001  222 2001:DB8:0:A10::/60
67.67.67.67     9         0X80000001  222 2001:DB8:0:A1B::/64
67.67.67.67     10        0X80000001  222 2001:DB8:0:A1C::/64
    # ADVERTISEMENTS:   6   CHECKSUM TOTAL: 0X000271C6
```

LS ORIGINATOR
The Router ID of the router that originated the advertisement.

LS ID
Uniquely identifies multiple external LSAs originated by the same router.

SEQNO, AGE
It is possible for several instances of an advertisement to be present in the IPv6 OSPF routing domain at any one time. However, only the most recent instance is kept in the IPv6 OSPF link state database (and printed by this command). The LS sequence number (Seqno) and LS age (Age) fields are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600. An asterisk (*) displayed beside an age value indicates that the DONOTAGE bit is on.

PREFIX
The prefix being described by the LSA.

At the end of the display, the total number of AS external advertisements is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement LS checksum fields. This information can be used to quickly determine whether two IPv6 OSPF routers have synchronized databases.

IPv6 OSPF area link state database:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,DATABASE,AREAID=*area-id* command displays the contents of a particular IPv6 OSPF area link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. Each advertisement is defined by the following three parameters:

- Its link state type (called Type)
- The advertising router (called the LS originator)
- Its link state ID

A sample output with an explanation of entries follows:

```
EZZ8126I IPV6 OSPF AREA LS DATABASE 829
              ROUTER LSAS (LS TYPE=2001)
LS ORIGINATOR   LS ID      SEQNO       AGE LINKS  RTR-TYPE
63.63.63.63     0          0X80000001  376 1
64.64.64.64     0          0X80000002  321 1      ABR,V
67.67.67.67     0          0X80000004  320 1      ABR,ASBR,V
68.68.68.68     0          0X80000002  595 1
   # ADVERTISEMENTS:    4   CHECKSUM TOTAL: 0X0001D024

              NETWORK LSAS (LS TYPE=2002)
LS ORIGINATOR   LS ID      SEQNO       AGE ROUTERS
68.68.68.68     14         0X80000004  375 4
   # ADVERTISEMENTS:    1   CHECKSUM TOTAL: 0X0000F5CC

              INTER-AREA PREFIX LSAS (LS TYPE=2003)
LS ORIGINATOR   LS ID      SEQNO       AGE PREFIX
64.64.64.64     4          0X80000002  395 2001:DB8:0:108::4/128
64.64.64.64     8          0X80000001  395 2001:DB8:0:108::2/128
64.64.64.64     9          0X80000001  395 2001:DB8:0:10::2/128
64.64.64.64     10         0X80000001  395 2001:DB8:0:10::/64
64.64.64.64     11         0X80000001  395 2:2:2:2:2:2:2:2/128
64.64.64.64     22         0X80000001  375 2001:DB8:0:120::4/128
64.64.64.64     26         0X80000001  321 2001:DB8:0:107::7/128
64.64.64.64     27         0X80000001  321 2001:DB8:0:120::7/128
64.64.64.64     28         0X80000001  321 2001:DB8:0:107::5/128
64.64.64.64     29         0X80000001  321 2001:DB8:0:20::5/128
64.64.64.64     30         0X80000001  321 2001:DB8:0:20::/64
67.67.67.67     15         0X80000002  358 2001:DB8:0:107::7/128
67.67.67.67     16         0X80000001  358 2:2:2:2:2:2:2:2/128
67.67.67.67     19         0X80000001  358 2001:DB8:0:107::5/128
67.67.67.67     20         0X80000001  358 2001:DB8:0:20::5/128
67.67.67.67     21         0X80000001  358 2001:DB8:0:20::/64
67.67.67.67     25         0X80000001  356 2001:DB8:0:120::7/128
67.67.67.67     26         0X80000001  317 2001:DB8:0:108::4/128
67.67.67.67     27         0X80000001  317 2001:DB8:0:108::2/128
67.67.67.67     28         0X80000001  317 2001:DB8:0:10::2/128
67.67.67.67     29         0X80000001  317 2001:DB8:0:10::/64
67.67.67.67     30         0X80000001  317 2001:DB8:0:120::4/128
   # ADVERTISEMENTS:   22   CHECKSUM TOTAL: 0X000E7320

              INTER-AREA ROUTER LSAS (LS TYPE=2004)
```

```
LS ORIGINATOR   LS ID      SEQNO       AGE DEST ROUTERID
64.64.64.64     3          0X80000001    8 62.62.62
67.67.67.67     2          0X80000001    9 62.62.62
   # ADVERTISEMENTS:    2   CHECKSUM TOTAL: 0X00007D88

              LINK LSAS (LS TYPE=0008)
LS ORIGINATOR   LS ID      SEQNO       AGE INTERFACE
63.63.63.63     34         0X80000001  387 NSQDIO1L6
64.64.64.64     16         0X80000001  402 NSQDIO1L6
67.67.67.67     20         0X80000002  640 NSQDIO1L6
68.68.68.68     14         0X80000002  638 NSQDIO1L6
   # ADVERTISEMENTS:    4   CHECKSUM TOTAL: 0X000295E4

              INTRA-AREA PREFIX LSAS (LS TYPE=2009)
LS ORIGINATOR   LS ID      SEQNO       AGE REF-LSTYPE REF-LSID
63.63.63.63     34         0X80000001  387 0X2001     0
63.63.63.63     36         0X80000001  387 0X2001     0
63.63.63.63     38         0X80000001  387 0X2001     0
64.64.64.64     16         0X80000001  402 0X2001     0
64.64.64.64     20         0X80000001  402 0X2001     0
67.67.67.67     20         0X80000002  639 0X2001     0
67.67.67.67     26         0X80000002  639 0X2001     0
68.68.68.68     14         0X80000003  595 0X2001     0
68.68.68.68     16         0X80000001 1738 0X2001     0
68.68.68.68     18         0X80000002  638 0X2001     0
68.68.68.68     65550      0X80000004  375 0X2002     14
   # ADVERTISEMENTS:   11   CHECKSUM TOTAL: 0X00068473
```

LS ORIGINATOR
The Router ID of the router that originated the advertisement.

LS ID
Uniquely identifies multiple LSAs of the same type originated by the same router.

SEQNO, AGE
It is possible for several instances of an advertisement to be present in the IPv6 OSPF routing domain at any one time. However, only the most recent instance is kept in the IPv6 OSPF link state database (and printed by this command). The LS sequence number (Seqno) and LS age (Age) fields are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600. An asterisk (*) displayed beside an age value indicates that the DONOTAGE bit is on.

LINKS
Number of links described by the LSA.

ROUTER TYPE
Indicates the level of function of the advertising router.

ROUTER TYPE	Description
ASBR	The router is an AS boundary router.
ABR	The router is an area border router.
V	The router is an endpoint of one of more fully adjacent virtual links having the described area as transit area.
W	The router is a wildcard multicast receiver (OMPROUTE will never set the W option on its own Router LSAs).

ROUTERS
The number of routers attached to the link described by the LSA.

PREFIX
The prefix being described by the LSA.

INTERFACE
Associated interface.

REF LS-TYPE,REF-LS ID
Identifies the referenced Router LSA or Network LSA.

At the end of each type of LSA in the display, the total number of advertisements of that type in the area database is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement LS checksum fields. This information can be used to quickly determine whether two IPv6 OSPF routers have synchronized databases.

IPv6 OSPF router routes:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,ROUTERS command displays all routes to other routers that have been calculated by IPv6 OSPF and are now present in the routing table. A sample output with an explanation of entries follows:

```
EZZ8125I IPV6 OSPF ROUTERS 820
DEST: 68.68.68.68
  NEXT HOP: FE80::8
  DTYPE:  RTR   RTYPE: SPF    COST: 1        AREA: 6.6.6.6
DEST: 64.64.64.64
  NEXT HOP: FE80::4
  DTYPE:   BR   RTYPE: SPF    COST: 1        AREA: 6.6.6.6
DEST: 65.65.65.65
  NEXT HOP: FE80::5:7
  DTYPE:  RTR   RTYPE: SPF    COST: 1        AREA: 0.0.0.0
DEST: 63.63.63.63
  NEXT HOP: FE80::3
  DTYPE:  RTR   RTYPE: SPF    COST: 1        AREA: 6.6.6.6
DEST: 62.62.62.62
  NEXT HOP: FE80::4
  DTYPE:  RTR   RTYPE: SPF    COST: 2        AREA: 0.0.0.0
DEST: 64.64.64.64
  NEXT HOP: FE80::4
  DTYPE:   BR   RTYPE: SPF    COST: 1        AREA: 0.0.0.0
```

DEST
Indicates the destination router's OSPF router ID.

NEXT HOP
Indicates the address of the next router on the path toward the destination host. A number in parentheses at the end of the address indicates the number of equal-cost routes to the destination.

DTYPE
Indicates the destination type:

ASBR
Indicates that the destination is an AS boundary router.

BR Indicates that the destination is an area border router.

FADD
Indicates a forwarding address (for external routes).

RTR
Indicates that the destination is a router.

RTYPE
Indicates the route type and how the route was derived:

SPF
Indicates that the route is an intra-area route (comes from the Dijkstra calculation).

SPIA
Indicates that it is an inter-area route (comes from considering Inter-Area Router advertisements).

COST
Displays the cost to reach the router.

AREA
Displays the OSPF area to which the destination router belongs.

IPv6 OSPF routing protocol statistics:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6OSPF,STATISTICS command displays statistics generated by the IPv6 OSPF routing protocol. (The keyword STATS can be substituted for STATISTICS.) The statistics indicate how well the implementation is performing, including its memory and network utilization. A sample output with an explanation of entries follows:

```
EZZ8124I IPV6 OSPF STATISTICS 839
ATTACHED AREAS:                  2  # DIJKSTRA RUNS:              12
OSPF PACKETS RCVD:             619  OSPF PACKETS RCVD W/ERRS:      0
TRANSIT NODES ALLOCATED:        26  TRANSIT NODES FREED:          17
LS ADV. ALLOCATED:             275  LS ADV. FREED:               175
QUEUE HEADERS ALLOC:            64  QUEUE HEADERS AVAIL:          64
INCREMENTAL SUMM. UPDATES:       5  INCREMENTAL VL UPDATES:        0
INCREMENTAL EXT. UPDATES:       27  PTRS TO INVALID LS ADV:        0
MULTICAST PKTS SENT:           421  UNICAST PKTS SENT:            40
LS ADV. AGED OUT:                0  LS ADV. FLUSHED:              41
```

ATTACHED AREAS
Indicates the number of areas to which the router has active interfaces.

# DIJKSTRA RUNS
Indicates how many times the IPv6 OSPF routing table has been calculated from scratch.

OSPF PACKETS RCVD
Covers all types of IPv6 OSPF protocol packets.

OSPF PACKETS RCVD W/ERRS
Indicates the number of IPv6 OSPF packets that have been received that were determined to contain errors.

TRANSIT NODES
Allocated to store Router LSAs and Network LSAs.

LS ADV
Allocated to store Inter-Area Prefix, Inter-Area Router, AS External, Link, and Intra-Area prefix LSAs.

QUEUE HEADERS
Form lists of link state advertisements. These lists are used in the flooding and database exchange processes. If the number of queue headers allocated is not equal to the number available, database synchronization with a neighbor is in progress.

INCREMENTAL SUMM UPDATES, INCREMENTAL VL UPDATES
Indicates how many times new Inter-Area Prefix or Inter-Area Router LSAs have caused the routing table to be partially rebuilt.

INCREMENTAL EXT. UPDATES
Displays the number of changes to external destinations that are incrementally installed in the routing table.

MULTICAST PKTS SENT
Covers IPv6 OSPF hello packets and packets sent during the flooding procedure.

UNICAST PKTS SENT
Covers IPv6 OSPF packet retransmissions and the Database Exchange procedure.

LS ADV. AGED OUT
Indicates the number of advertisements that have hit 60 minutes. Link state advertisements are aged out after 60 minutes. Usually they are refreshed before this time.

LS ADV. FLUSHED
Indicates the number of advertisements removed (and not replaced) from the link state database.

Examples using the IPV6RIP command:

All IPv6 RIP information:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6RIP,ALL command lists all IPv6 RIP-related information. A sample output with an explanation of entries follows:

```
EZZ8030I IPV6 RIP CONFIGURATION
TRACE6: 1, DEBUG6: 0
STACK AFFINITY:  TCPCS6
IPV6 RIP: ENABLED
IPV6 RIP DEFAULT ORIGINATION: ALWAYS, COST = 1

EZZ8027I IPV6 RIP INTERFACES
                                    ---------SEND-----------  --RCV--
NAME               MTU STATE IN OUT PRF HST STA DEF RADV PSN  PRF HST
NSQDIO3L6         9000    UP  1   0  NO YES YES YES   NO  NO  YES YES
LOSAFE3           4000   N/A  1   0 YES  NO YES  NO  YES YES  YES  NO


EZZ8031I IPV6 RIP ROUTE ACCEPTANCE
ACCEPT IPV6 RIP UPDATES ALWAYS FOR:
  2001:DB8::1:9:67:115:66
  2001:DB8:0:0:A1B::

EZZ8029I GLOBAL IPV6 RIP FILTERS

SEND ONLY: VIRTUAL, DEFAULT

IGNORE IPV6 RIP UPDATES FROM:
  FE80::1:2:3:4

FILTERS: NOSEND     2001:DB8::1:8:E2:43:28/128
         NORECEIVE  2001:DB8:0:0:A1E::/64
```

TRACE6
Displays the level of tracing currently in use by OMPROUTE IPv6 routing protocols.

DEBUG6
Displays the level of debugging currently in use by OMPROUTE IPv6 routing protocols.

STACK AFFINITY
Displays the name of the stack on which OMPROUTE is running.

IPV6 RIP DEFAULT ORIGINATION
Indicates the conditions under which IPv6 RIP supports default route generation and the advertised cost for the default route.

The remainder of the TCPIP,*tcpipjobname*,OMPROUTE,IPV6RIP,ALL output is described in the following sections.

IPv6 RIP routes to be accepted:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6RIP,ACCEPTED command lists the routes to be unconditionally accepted, as configured with the IPV6_ACCEPT_RIP_ROUTE statement. A sample output follows:

```
EZZ8030I IPV6 RIP ROUTE ACCEPTANCE
ACCEPT IPV6 RIP UPDATES ALWAYS FOR:
2001:DB8::1:0009:0067:0115:0066
2001:DB8::A1B::
```

ACCEPT IPV6 RIP UPDATES ALWAYS FOR
Indicates the prefixes and hosts for which updates are always accepted.

IPv6 RIP interface statistics and parameters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6RIP,INTERFACE,NAME=*if_name* command displays statistics and parameters related to IPv6 RIP interfaces. (The keyword IF can be substituted for INTERFACE.) If no NAME= parameter is given (DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6RIP,INTERFACE), a single line is printed summarizing each interface. (See example 1.) If a NAME= parameter is given, detailed statistics for that interface are displayed. (See example 2.)

```
----  Example 1  ----
EZZ8027I IPV6 RIP INTERFACES
                                    ---------SEND-----------  --RCV--
NAME               MTU STATE IN OUT PRF HST STA DEF RADV PSN  PRF HST
NSQDIO3L6         9000    UP  1   0  NO YES YES YES   NO  NO  YES YES
LOSAFE3           4000   N/A  1   0 YES  NO YES  NO  YES YES  YES  NO
```

NAME
Indicates the name of the IPv6 RIP interface.

MTU
Indicates the value of the maximum transmission unit learned from the TCP/IP stack for the interface.

STATE
Indicates the status of the interface. Values are:

UP The interface is up.

DOWN
The interface is known to TCP/IP but is down.

N/A
The interface is defined to OMPROUTE, but the TCP/IP stack has not informed OMPROUTE that the interface is installed. For detailed interface status information, use the DISPLAY TCPIP,*procname*,NETSTAT,DEVLINKS command.

IN Specifies the value of the metric to be added to IPv6 RIP routes received over this interface.

OUT
Specifies the value of the metric to be added to IPv6 RIP routes advertised over this interface.

SEND

PRF
Indicates whether prefix routes are advertised in IPv6 RIP responses sent over this interface.

HST
Indicates whether host routes are advertised in IPv6 RIP responses sent over this interface.

STA
Indicates whether static routes are advertised in IPv6 RIP responses sent over this interface.

DEF
Indicates whether the default route, if available, is advertised in IPv6 RIP responses sent over this interface.

RADV
Indicates whether router advertisement routes are advertised in IPv6 RIP responses sent over this interface.

PSN
Indicates whether poisoned reverse routes are advertised in IPv6 RIP responses sent over this interface. A poisoned reverse route is one with an infinite metric (a metric of 16).

RECEIVE

PRF
Indicates whether prefix routes are accepted in IPv6 RIP responses received over this interface.

HST
Indicates whether host routes are accepted in IPv6 RIP responses received over this interface.

```
----  Example 2  ----
EZZ8028I IPV6 RIP INTERFACE DETAILS
INTERFACE NAME:     LOSAFE6
INTERFACE ADDRESS:  FE80::1:2:3:1
                    2001:DB8::1:2:3:1
NTERFACE PREFIX:   RADV 12AB::/16
                    STAT 9800:1234::/32
MTU:                    2000     STATE:                  UP
IN METRIC:              1        OUT METRIC:             0
SEND PREFIX ROUTES:     YES      SEND HOST ROUTES:       NO
SEND STATIC ROUTES:     NO       SEND DEFAULT ROUTES:    NO
SEND RTR. ADV. ROUTES:  YES      SEND POIS. REV. ROUTES: NO
RECEIVE PREFIX ROUTES:  YES      RECEIVE HOST ROUTES:    YES

SEND ONLY:  VIRTUAL, DEFAULT

FILTERS: SEND       2001:DB8::1:8:E2:43:28/128
         NORECEIVE  2001:DB8::A1E::/64
```

INTERFACE NAME
Indicates the interface name.

INTERFACE ADDRESS
Indicates the IP addresses that have been learned from the TCP/IP stack for the interface.

INTERFACE PREFIX
Lists the interface prefixes. RADV indicates the prefix was learned through

IPv6 Router Discovery. STAT indicates it was statically defined to this interface using the PREFIX parameter of the IPV6_RIP_INTERFACE statement.

MTU
Indicates the value of the maximum transmission unit learned from the TCP/IP stack for the interface.

STATE
Indicates the status of the interface. Values are:

UP The interface is up.

DOWN
The interface is known to TCP/IP but is down.

N/A
The interface is defined to OMPROUTE, but the TCP/IP stack has not informed OMPROUTE that the interface is installed. For detailed interface status information, use the DISPLAY TCPIP,*procname*,NETSTAT,DEVLINKS command.

IGNORED
The interface is known to TCP/IP but is being ignored by OMPROUTE.

IN METRIC
Specifies the value of the metric to be added to IPv6 RIP routes received over this interface.

OUT METRIC
Specifies the value of the metric to be added to IPv6 RIP routes advertised over this interface.

SEND PREFIX ROUTES
Indicates whether prefix routes are advertised in IPv6 RIP responses sent over this interface.

SEND HOST ROUTES
Indicates whether host routes are advertised in IPv6 RIP responses sent over this interface.

SEND STATIC ROUTES
Indicates whether static routes are advertised in IPv6 RIP responses sent over this interface.

SEND DEFAULT ROUTES
Indicates whether the default route, if available, is advertised in IPv6 RIP responses sent over this interface.

SEND RTR. ADV. ROUTES
Indicates whether router advertisement routes are advertised in IPv6 RIP responses sent over this interface.

SEND POIS. REV. ROUTES
Indicates whether poisoned reverse routes are advertised in IPv6 RIP responses sent over this interface. A poisoned reverse route is one with an infinite metric (a metric of 16).

RECEIVE PREFIX ROUTES
Indicates whether prefix routes are accepted in IPv6 RIP responses received over this interface.

RECEIVE HOST ROUTES
Indicates whether host routes are accepted in IPv6 RIP responses received over this interface.

SEND ONLY
Indicates the route-type restrictions on IPv6 RIP sends for this interface.

FILTERS
Indicates the send and receive filters for this interface.

Global IPv6 RIP filters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,IPV6RIP,FILTERS command displays the Global IPv6 RIP filters. A sample output with an explanation of entries follows:

```
EZZ8029I GLOBAL IPV6 RIP FILTERS

SEND ONLY: VIRTUAL, DEFAULT

IGNORE IPV6 RIP UPDATES FROM:
  FE80::1:2:3:4

FILTERS: NOSEND     2001:DB8::1:8:E2:43:28/128
         NORECEIVE  2001:DB8::A1E::/64
```

SEND ONLY
Indicates the global route-type restrictions on IPv6 RIP sends that apply to all IPv6 RIP interfaces.

IGNORE IPV6 RIP UPDATES FROM
Indicates the IPv6 RIP routers from which advertisements will not be accepted.

FILTERS
Indicates the global send and receive filters that apply to all IPv6 RIP interfaces.

Examples using the GENERIC6 command:

All IPv6 generic information:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,GENERIC6,ALL command lists all IPv6 generic information, which is information that is not specific to a routing protocol. A sample output with an explanation of entries follows:

```
EZZ8053I IPV6 GENERIC CONFIGURATION 067
TRACE6: 2, DEBUG6: 3
IPV6 TRACE DESTINATION: /TMP/6MPROUT3.DBG
STACK AFFINITY: TCPCS3

EZZ8060I IPV6 GENERIC INTERFACES
NAME               MTU STATE CONFIGURED
MPCPTPV66        65535    UP      NO
GENERIC_INTF      1280   N/A     YES
```

TRACE6
Displays the level of tracing currently in use by OMPROUTE IPv6 routing protocols.

DEBUG6
Displays the level of debugging currently in use by OMPROUTE IPv6 routing protocols.

IPV6 TRACE DESTINATION
Displays the file name of the IPv6 trace destination, or OMPCTRC if that destination is the OMPROUTE CTRACE.

Restriction: The trace destination is displayed in upper case on the console, regardless of the case of the actual case-sensitive file name, if the destination is a z/OS UNIX file.

STACK AFFINITY
Displays the name of the stack on which OMPROUTE is running.

The remainder of the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,GENERIC6,ALL output is described in the following sections.

IPv6 generic interface statistics and parameters:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,GENERIC6,INTERFACE,NAME=*if-name* command displays statistics and parameters related to IPv6 generic interfaces. (The keyword IF can be substituted for INTERFACE.) If no NAME= parameter is given (DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,GENERIC6,INTERFACE), a single line is printed summarizing each interface. (See Example 1.) If a NAME= parameter is given, detailed statistics for that interface are displayed. (See Example 2.)

```
----  Example 1  ----

EZZ8060I IPV6 GENERIC INTERFACES
NAME               MTU STATE CONFIGURED
MPCPTPV66        65535    UP      NO
GENERIC_INTF      1280   N/A     YES
```

NAME
Indicates the name of the IPv6 generic interface.

MTU
Indicates the value of the maximum transmission unit learned from the TCP/IP stack for the interface.

STATE
Indicates the status of the interface. Values are:

UP The interface is up.

DOWN
The interface is known to TCP/IP but is down.

N/A
The interface is defined to OMPROUTE, but the TCP/IP stack has not informed OMPROUTE that the interface is installed. For detailed interface status information, use the DISPLAY TCPIP,*procname*,NETSTAT,DEVLINKS command.

IGNR
The interface is known to TCP/IP but is being ignored by OMPROUTE.

CONFIGURED
Indicates whether or not the interface was configured to OMPROUTE.

```
---- Example 2  ----
EZZ8065I IPV6 GENERIC INTERFACE DETAILS
INTERFACE NAME:     LOSAFE6
INTERFACE ADDRESS:  FE80::9:9:9:8
                    2001:DB8::9:9:9:8
INTERFACE PREFIX:   RADV 1201::/16
                    STAT 9801:4321::/32

MTU:                    2000
STATE:                  UP
CONFIGURED:             YES
```

INTERFACE NAME
Indicates the interface name.

INTERFACE ADDRESS
Indicates the IP addresses that have been learned from the TCP/IP stack for the interface.

INTERFACE PREFIX
Lists the interface prefixes. RADV indicates the prefix was learned using IPv6 Router Discovery. STAT indicates it was statically defined to this interface using the PREFIX parameter of the IPV6_INTERFACE statement.

MTU
Indicates the value of the maximum transmission unit learned from the TCP/IP stack for the interface.

STATE
Indicates the status of the interface. Values are:

UP The interface is up.

DOWN
The interface is known to TCP/IP but is down.

N/A
The interface is defined to OMPROUTE, but the TCP/IP stack has not informed OMPROUTE that the interface is installed. For detailed interface status information use the DISPLAY TCPIP,*procname*,NETSTAT,DEVLINKS command.

IGNR
The interface is known to TCP/IP but is being ignored by OMPROUTE.

CONFIGURED
Indicates whether or not the interface was configured to OMPROUTE.

Examples using the RT6TABLE command:

The following sections show the examples of using the RT6TABLE command.

OMPROUTE IPv6 main routing table:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE command displays all of the routes in the OMPROUTE IPv6 main routing table. A sample output with an explanation of entries follows.

Result: This command displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.

```
EZZ7979I IPV6 ROUTING TABLE 641
DESTINATION: 4:4:4:4:4:4:4:4/128
  NEXT HOP: FE80::4
  TYPE: SPF           COST:  1         AGE: 2170
DESTINATION: 6:6:6:6:6:6:6:6/128
  NEXT HOP: FE80::6:7
  TYPE: RIP           COST:  2         AGE: 0
DESTINATION: 7:7:7:7:7:7:7:7/128
  NEXT HOP: ::
  TYPE: SPF *         COST:  0         AGE: 59
DESTINATION: 2001:DB8:0:10::/64
  NEXT HOP: FE80::4
  TYPE: SPF           COST:  3         AGE: 32
DESTINATION: 2001:DB8:0:103::6/128
  NEXT HOP: FE80::6:7
  TYPE: RIP           COST:  2        AGE: 0
DESTINATION: 2001:DB8:0:103::7/128
  NEXT HOP: ::
  TYPE: DIR *         COST:  1        AGE: 2209
DESTINATION: 2001:DB8:0:108::2/128
  NEXT HOP: FE80::4
  TYPE: SPF           COST:  2        AGE: 32
DESTINATION: 2001:DB8:0:108::4/128
  NEXT HOP: FE80::4
  TYPE: SPF           COST:  1        AGE: 32
DESTINATION: 2001:DB8:0:120::/64
  NEXT HOP: ::
  TYPE: SPF *         COST:  1        AGE: 2172
DESTINATION: 2001:DB8:0:120::4/128
  NEXT HOP: FE80::4
  TYPE: SPF           COST:  1        AGE: 2170
DESTINATION: 2001:DB8:0:120::7/128
  NEXT HOP: ::
  TYPE: SPF *         COST:  0        AGE: 2172
DESTINATION: 2001:DB8:0:A10::/60
  NEXT HOP: FE80::6:7
  TYPE: RIP           COST:  2        AGE: 0
DESTINATION: 2001:DB8:0:A1B::/64
  NEXT HOP: FE80::6:7
  TYPE: RIP           COST:  2        AGE: 0
DESTINATION: 2001:DB8:0:A1C::/64
  NEXT HOP: FE80::6:7
  TYPE: RIP           COST:  2        AGE: 0
                      0 NETS DELETED, 5 NETS INACTIVE
```

DESTINATION
Indicates the IP destination, along with its prefix length.

NEXT HOP
Indicates the IP address of the next router on the path toward the destination. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. Use the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE,DEST=*ip_addr* command to obtain a list of the next hops.

TYPE
Indicates how the route was derived:

DFLT
Indicates a route defined using the IPV6_DEFAULT_ROUTE configuration statement in the OMPROUTE configuration file.

DIR
Indicates a directly connected prefix or host.

RIP
Indicates a route that was learned through the IPv6 RIP protocol.

DEL
Indicates the route has been deleted.

Restriction: Deleted routes are shown only when RIP is active and only as long as RIP needs to advertise to neighboring routers that they have been deleted.

STAT
Indicates a nonreplaceable statically configured route.

SPF
Indicates that the route is an IPv6 OSPF intra-area route.

SPIA
Indicates that the route is an IPv6 OSPF interarea route.

SPE1
Indicates IPv6 OSPF external routes (type 1).

SPE2
Indicates IPv6 OSPF external routes (type 2).

RANGE
Indicates a route type that is an active IPv6 OSPF area address range and is not used in forwarding packets.

RSTA
Indicates a static route that is defined as replaceable.

RADV
Indicates a route that was learned by the TCP/IP stack through the IPv6 Router Discovery protocol.

An asterisk (*) after the route type indicates that the route has a directly connected backup. A percent sign (%) after the route type indicates that IPv6 RIP updates are always accepted for this destination.

COST
Indicates the route cost.

Table 5. OMPROUTE IPv6 Route Type and COST Value mapping

Route Type	COST Value
SPF or SPIA	The OSPF cost of the route.
SPE1	The OSPF cost to get to the AS boundary router or forwarding address that is used to reach the destination, plus the external cost.
SPE2	The external cost.
RIP	The RIP metric
STAT or RSTA	• 0 when the route is direct. • 1 when the route is indirect.
DIR or SBNT	1
RNGE	The OSPF cost of the range.
DFLT	0
RADV	• 1 when the router advertisement indicated a preference of high. • 2 when the router advertisement indicated a preference of medium. • 3 when the router advertisement indicated a preference of low.

AGE
Indicates the time that has elapsed since the routing table entry was last refreshed. For routes that have the route type DEL or RIP, this value increments by a factor of 10 for each 10–second increase in age. If the TCP/IP stack enters a storage shortage condition, all routes that have the route type DEL or RIP are refreshed when the shortage condition is relieved.

NETS DELETED
Indicates the number of routes that have been deleted from the OMPROUTE routing table and not replaced. Use the DTCPIP,,OMPROUTE,RT6TABLE,DELETED command to list these routes.

NETS INACTIVE
Used for internal debugging purposes only.

Route expansion information for the OMPROUTE IPv6 main routing table:
Use the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE,DEST=*ip_addr* command to obtain information about a particular route in the OMPROUTE IPv6 main routing table. When multiple equal-cost routes exist, use this command to obtain a list of the next hops. A sample output with an explanation of entries follows:

```
EZZ7980I IPV6 ROUTE EXPANSION
DESTINATION: 2001:DB8::9:67:115:13/128
ROUTE TYPE:  RIP
COST:        5
AGE:         231
NEXT HOP(S): FE80::7:7:7:7                          (LOSAFE6)
             FE80::8:8:8:8                          (LOSAFE6)
             FE80::9:9:9:9                          (LOSAFE3)
```

DESTINATION
Indicates the IP destination, along with its prefix length.

ROUTE TYPE
Indicates how the route was derived:

DFLT
Indicates a route defined using the IPV6_DEFAULT_ROUTE configuration statement in the OMPROUTE configuration file.

DIR
Indicates a directly connected prefix or host.

RIP
Indicates a route that was learned through the IPv6 RIP protocol.

STAT
Indicates a nonreplaceable statically configured route.

SPF
Indicates that the route is an IPv6 OSPF intra-area route.

SPIA
Indicates that the route is an IPv6 OSPF interarea route.

SPE1
Indicates IPv6 OSPF external routes (type 1).

SPE2
Indicates IPv6 OSPF external routes (type 2).

RANGE
Indicates a route type that is an active IPv6 OSPF area address range and is not used in forwarding packets.

RSTA
Indicates a static route that is defined as replaceable.

RADV
Indicates a route that was learned by the TCP/IP stack through the IPv6 Router Discovery protocol.

An asterisk (*) after the route type indicates that the route has a directly connected backup. A percent sign (%) after the route type indicates that IPv6 RIP updates are always accepted for this destination.

COST
Indicates the route cost.

Table 6. IPv6 Route Type and COST Value mapping

Route Type	COST Value
SPF or SPIA	The OSPF cost of the route.
SPE1	The OSPF cost to get to the AS boundary router or forwarding address that is used to reach the destination, plus the external cost.
SPE2	The external cost.
RIP	The RIP metric.
STAT or RSTA	• 0 when the route is direct • 1 when the route is indirect
DIR or SBNT	1
RNGE	The OSPF cost of the range.
DFLT	0
RADV	• 1 when the router advertisement indicated a preference of high. • 2 when the router advertisement indicated a preference of medium. • 3 when the router advertisement indicated a preference of low.

AGE
Indicates the time that has elapsed since the routing table entry was last refreshed. For routes that have the route type DEL or RIP, this value increments by a factor of 10 for each 10 second increase in age. If the TCP/IP stack enters a storage shortage condition, all routes that have the route type DEL or RIP are refreshed when the shortage condition is relieved.

NEXT HOP(S)
Indicates the IP address of the next router and the interface used to reach that router for each of the paths toward the destination.

All OMPROUTE IPv6 policy-based routing tables:
The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,RT6TABLE,PRTABLE=ALL` command displays all of the routes in all of the OMPROUTE IPv6 policy-based routing tables. The dynamic routing parameters that are configured to the policy agent for each table are displayed following the routes for that table. A sample output with an explanation of the entries follows.

Results:

- This command displays the contents of the working tables that OMPROUTE uses; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.
- If a policy-based routing table is configured with no IPv6 dynamic routing parameters, OMPROUTE has no knowledge of that routing table for IPv6. The routing table is not included in the display of OMPROUTE IPv6 policy-based routing tables.

```
EZZ7979I IPV6 ROUTING TABLE 214
TABLE NAME: SECLOW2
DESTINATION: 6:6:6:6:6:6:6:6/128
  NEXT HOP: FE80::6:7
  TYPE:  RIP          COST:  2         AGE: 10
```

```
DESTINATION: 2001:DB8:0:103::6/128
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:103::7/128
  NEXT HOP: ::
  TYPE:  DIR*          COST:  1        AGE: 66
DESTINATION: 2001:DB8:0:A10::/60
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:A1B::/64
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:A1C::/64
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:C1C::/64
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
                       0 NETS DELETED
DYNAMIC ROUTING PARAMETERS
  INTERFACE:  MPCPTP7TO6       NEXT HOP: ANY

TABLE NAME: SECLOW1
DESTINATION: 4:4:4:4:4:4:4:4/128
  NEXT HOP: FE80::4 (2)
  TYPE:  SPF           COST:  1        AGE: 65
DESTINATION: 2001:DB8:0:10::/64
  NEXT HOP: FE80::4 (2)
  TYPE: SPF            COST:  3        AGE: 65
DESTINATION: 2001:DB8:0:108::2/128
  NEXT HOP: FE80::4 (2)
  TYPE: SPF            COST:  2        AGE: 65
DESTINATION: 2001:DB8:0:108::4/128
  NEXT HOP: FE80::4 (2)
  TYPE: SPF            COST:  1        AGE: 65
DESTINATION: 2001:DB8:0:120::/64
  NEXT HOP: ::
  TYPE:  SPF*          COST:  1        AGE: 65
DESTINATION: 2001:DB8:0:120::4/128
  NEXT HOP: FE80::4 (2)
  TYPE:  SPF           COST:  1        AGE: 65
DESTINATION: 2001:DB8:0:120::7/128
  NEXT HOP: ::
  TYPE:  SPF*          COST:  0        AGE: 65
DESTINATION: 2001:DB8:0:C1C::/64
  NEXT HOP: FE80::4 (2)
  TYPE:  SPF           COST:  3        AGE: 65
                       0 NETS DELETED
DYNAMIC ROUTING PARAMETERS
  INTERFACE:  NSQDIO1L6        NEXT HOP: FE80::4
```

TABLE NAME

Indicates the name of the policy-based routing table.

INTERFACE

Indicates the name of an interface that is specified in a dynamic routing parameter for the policy-based routing table. If the interface is not currently defined to the TCP/IP stack as an IPv6 interface or the interface is inactive on the TCP/IP stack, the name is preceded by an asterisk (*).

NEXT HOP

Indicates the next hop router IP address that is specified in a dynamic routing parameter for the policy-based routing table. The value ANY is displayed when no next-hop router IP address is specified for the dynamic routing parameter.

See “OMPROUTE IPv6 main routing table” on page 94 for additional field descriptions.

OMPROUTE IPv6 policy-based routing table:
The **DISPLAY TCPIP,tcpipjobname,OMPROUTE,RT6TABLE,PRTABLE=prname** command displays all of the routes in a single OMPROUTE IPv6 policy-based routing table. The dynamic routing parameters that are configured to the policy agent for the table are displayed following the routes for the table. A sample output with explanation of entries follows.

Results:

- This command displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.
- If a policy-based route table is configured with no IPv6 dynamic routing parameters, OMPROUTE has no knowledge of that route table for IPv6. You cannot use that route table with this command.

```
EZZ7979I IPV6 ROUTING TABLE 214
TABLE NAME: SECLOW2
DESTINATION: 6:6:6:6:6:6:6:6/128
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:103::6/128
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:103::7/128
  NEXT HOP: ::
  TYPE:  DIR*          COST:  1        AGE: 66
DESTINATION: 2001:DB8:0:A10::/60
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:A1B::/64
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:A1C::/64
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
DESTINATION: 2001:DB8:0:C1C::/64
  NEXT HOP: FE80::6:7
  TYPE:  RIP           COST:  2        AGE: 10
                       0 NETS DELETED
DYNAMIC ROUTING PARAMETERS
  INTERFACE:  MPCPTP7TO6       NEXT HOP: ANY
```

See “All OMPROUTE IPv6 policy-based routing tables” on page 98 for field descriptions.

Route expansion information for OMPROUTE IPv6 policy-based routing table:
Use the **DISPLAY TCPIP,tcpipjobname,OMPROUTE,RT6TABLE,PRTABLE=prname,DEST=ip-addr** command to obtain information about a particular route in an OMPROUTE IPv6 policy-based routing table. When multiple equal-cost routes exist, use this command to obtain a list of the next hops. A sample output with explanation of entries follows.

Results:

- This command displays information from the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The OMPROUTE routing table might contain information that is different from the information in the TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see "DISPLAY TCPIP,,NETSTAT" on page 9.

- If a policy-based route table is configured with no IPv6 dynamic routing parameters, OMPROUTE has no knowledge of that route table for IPv6. You cannot use that route table with this command.

```
EZZ7980I IPV6 ROUTE EXPANSION 384
TABLE NAME: SECLOW1
DESTINATION: 4:4:4:4:4:4:4:4/128
ROUTE TYPE:  SPF
COST:        1
AGE:         963
NEXT HOP(S): FE80::4                               (NSQDIO1L6)
             FE80::9:67:4:4                        (EZ6SAMEMVS)
```

TABLE NAME
Indicates the name of the policy-based routing table.

See "Route expansion information for the OMPROUTE IPv6 main routing table" on page 97 for additional field descriptions.

Route expansion information for all OMPROUTE IPv6 policy-based routing tables:
Use the **DISPLAY TCPIP,tcpipjobname,OMPROUTE,RT6TABLE,PRTABLE=ALL,DEST=ip-addr** command to obtain information from all of the OMPROUTE IPv6 policy-based routing tables about a particular route. When multiple equal-cost routes exist in a table, use this command to obtain a list of the next hops. A sample output with explanation of entries follows.

Results:

- This command displays information from the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see "DISPLAY TCPIP,,NETSTAT" on page 9.
- If a policy-based route table is configured with no IPv6 dynamic routing parameters, OMPROUTE has no knowledge of that route table for IPv6. The route table does not appear in the display of OMPROUTE IPv6 route tables.

```
EZZ7980I IPV6 ROUTE EXPANSION 384
TABLE NAME: SECLOW1
DESTINATION: 4:4:4:4:4:4:4:4/128
ROUTE TYPE:  SPF
COST:        1
AGE:         963
NEXT HOP(S): FE80::4                               (NSQDIO1L6)
             FE80::9:67:4:4                        (EZ6SAMEMVS)
```

TABLE NAME
Indicates the name of the policy-based routing table.

See "Route expansion information for the OMPROUTE IPv6 main routing table" on page 97 for additional field descriptions.

Deleted OMPROUTE IPv6 routes:
The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE,DELETED command displays the routes that have been deleted from the OMPROUTE IPv6 routing table and that have not been replaced or recycled through garbage collection (garbage collection occurs only when IPv6 RIP is running). A sample output

follows. The explanation for the entries is the same as for the Display TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE command (see "OMPROUTE IPv6 main routing table" on page 94).

```
D TCPIP,TCPCS6,OMPROUTE,RT6TABLE,DELETED
         EZZ7979I IPV6 DELETED ROUTES 593
         DESTINATION: 2001:DB8:10::11:2:1/128
           NEXT HOP: ::
           TYPE:  DEL         COST:  1        AGE: 76484
         DESTINATION: 2001:DB8:10::12:2:1/128
           NEXT HOP: ::
           TYPE:  DEL         COST:  1        AGE: 76484
         DESTINATION: 2001:DB8:10::81:1:1/128
           NEXT HOP: ::
           TYPE:  DEL         COST:  1        AGE: 76506
         DESTINATION: 2001:DB8:10::87:1:1/128
           NEXT HOP: ::
           TYPE:  DEL         COST:  1        AGE: 76506
         DESTINATION: 2001:DB8:10::91:1:1/128
           NEXT HOP: ::
           TYPE:  DEL         COST:  1        AGE: 76506
```

DISPLAY TCPIP,,OSAINFO

Use the DISPLAY TCPIP,,OSAINFO command to retrieve information for active IPAQENET and IPAQENET6 interfaces. An interface represents a single datapath device of an OSA-Express feature. The information is retrieved directly from the OSA-Express feature.

Tips:

- If you have an INTERFACE and a DEVICE or LINK definition with the same port name and both are active, specifying either the INTERFACE or link name on the command will generate a report with both IPv4 and IPv6 information.
- You can use the DISPLAY TCPIP,,OSAINFO command to retrieve information for an active OSAENTA interface although only the base portion of the report is pertinent.

The command output provides the following sections of information:

Base Contains physical characteristics and attributes for the interface and OSA-Express feature.

Registered addresses
Contains the Layer 2 MAC addresses or Layer 3 unicast and multicast IPv4 and IPv6 addresses registered to the OSA-Express feature.

QDIO inbound workload queueing routing variables
If QDIO inbound workload queueing is in effect for the interface, this section contains the routing variables for the ancillary input queues. Routing variables identify which inbound packets are to be presented on an ancillary input queue. For more information about ancillary input queues, see QDIO inbound workload queueing in z/OS Communications Server: IP Configuration Guide.

Both the modifiers and the MAX parameters can be used to limit the number of output lines.

Format:

```
►►──Display ──TCPIP──,──┬────────────┬──,──OSAinfo──,──INTFName=──intf_name────────►
                        └─procname───┘
```

```
   ┌──────────────┐
   ▼              │  (1)  ┌─,MAX=200──┐
►──┬──────────────┴───────┼───────────┼────────────────────────────────────►◄
   ├─,BASE────────┤       ├─,MAX=*────┤
   ├─,BULKdata────┤       └─,MAX=lines┘
   ├─,EE──────────┤
   ├─,REGAddrs────┤
   └─,SYSDist─────┘
```

Notes:

1 If no modifiers are specified, all sections for which information exists are displayed.

Rule: You must specify the parameters in the order that the syntax diagram shows.

Parameters:

OSAINFO
Requests OSA information.

INTFNAME = *intf_name*
Specifies the name of the OSA-Express QDIO interface whose datapath device information is requested. The *intf_name* value can be one of the following names:

- The name that was configured on a LINK IPAQENET profile statement.
- The name that was configured on an INTERFACE IPAQENET or IPAQENET6 profile statement.
- The name of an OSAENTA trace interface, which is EZANTA*portname*, where the *portname* value is the name that is specified on the PORTNAME keyword in the TRLE for the OSA-Express port that is being traced.

Tip: To obtain a list of names to use as the value of the INTFNAME parameter, use the Netstat DEvlinks/**-d** command.

BASE
Indicates that the physical characteristics and attributes of the interface and OSA-Express feature are to be included in the report.

BULKDATA
Indicates that QDIO inbound workload queueing routing variables for the BULKDATA ancillary queue are to be included in the report. The BULKDATA routing variables are comprised of source and destination IP addresses, source and destination ports, and protocol. That combination uniquely identifies those packets that the OSA-Express will route to the BULKDATA ancillary queue.

EE Indicates that QDIO inbound workload queueing routing variables for the Enterprise Extender (EE) ancillary queue are to be included in the report. The EE routing variables are comprised of destination IP addresses, destination ports, and protocol. That combination uniquely identifies those packets that the OSA-Express will route to the EE ancillary queue.

REGADDRS
Indicates that registered Layer 2 MAC addresses or Layer 3 unicast and multicast addresses are to be included in the report.

SYSDIST
Indicates that QDIO inbound workload queueing routing variables for the SYSDIST ancillary queue are to be included in the report. The SYSDIST routing

variables are comprised of destination IP addresses and protocol. That combination uniquely identifies those packets that the OSA-Express will route to the SYSDIST ancillary queue.

MAX = *lines*| *

The maximum number of lines to be displayed on the console. Valid *lines* values are in the range 4 - 65533. Specify an asterisk (*) to allow up to 65533 lines to be displayed.

- If MAX=* is specified and the report is truncated as the result of exceeding the multi-line WTO maximum, the following message is displayed:

  ```
  Report truncated due to greater than 65533 lines of output
  ```

- In all other cases, the total number of lines that is displayed and the total number of lines that could have been displayed are shown in the following output line, where *n* is the number of lines displayed and *m* is the total number of lines that could have been displayed.

  ```
  n of m lines displayed
  ```

Examples:

```
DISPLAY TCPIP,,OSAINFO,INTFNAME=LNK29D,MAX=500
```

Example of IPv4 interface reply:

```
EZD0031I TCP/IP CS V2R1  TCPIP Name: TCPCS     15:14:15
Display OSAINFO results for IntfName: LNK29D
PortName: DEV29D    PortNum: 01  Datapath: 3902   RealAddr: 0002
PCHID: 0451         CHPID: 29    CHPID Type: OSD  OSA code level: 6760
Gen: OSA-E3         Active speed/mode: 1000 mb/sec full duplex
Media: Singlemode Fiber        Jumbo frames: Yes  Isolate: No
PhysicalMACAddr: 643B88F30000  LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 3   Ancillary queues in use: 2
Connection Mode: Layer 3       IPv4: Yes  IPv6: No
SAPSup: 00010293               SAPEna: 00010293
IPv4 attributes:
  VLAN ID: N/A             VMAC Active: No
  Defined Router: Non      Active Router: No
  AsstParmsEna: 00215C66   OutCkSumEna: 00000000  InCkSumEna: 00000000
Registered Addresses:
  IPv4 Unicast Addresses:
    ARP: Yes  Addr: 10.10.10.10
    Total number of IPv4 addresses:      1
  IPv4 Multicast Addresses:
    MAC: 01005E000001  Addr: 224.0.0.1
    Total number of IPv4 addresses:      1
Ancillary Input Queue Routing Variables:
  Queue Type: BULKDATA  Queue ID:  2  Protocol: TCP
    Src: 11.1.1.11..100
    Dst: 12.12.12.12..100
    Src: 13.3.3.13..101
    Dst: 14.14.14.14..101
    Total number of IPv4 connections:      2
  Queue Type: SYSDIST   Queue ID:  3  Protocol: TCP
    Addr: 10.10.10.10
    Total number of IPv4 addresses:      1
33 OF 33 Lines Displayed
End of report
```

Example of IPv6 interface reply:

```
EZD0031I TCP/IP CS V2R1  TCPIP Name: TCPCS     15:14:15
Display OSAINFO results for IntfName: LNK29D
PortName: DEV29D    PortNum: 01  Datapath: 3902   RealAddr: 0002
PCHID: 0451         CHPID: 29    CHPID Type: OSD  OSA code level: 6760
Gen: OSA-E3         Active speed/mode: 1000 mb/sec full duplex
Media: Singlemode Fiber        Jumbo frames: Yes  Isolate: No
```

```
PhysicalMACAddr: 643B88F30000  LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 3   Ancillary queues in use: 2
Connection Mode: Layer 3       IPv4: No   IPv6: Yes
SAPSup: 00010293               SAPEna: 00010293
IPv6 attributes:
  VLAN ID: N/A             VMAC Active: Yes
  VMAC Addr: 643B88F30001  VMAC Origin: Cfg       VMAC Router: All
  AsstParmsEna: 00215C66   OutCkSumEna: 00000000  InCkSumEna: 00000000
Registered Addresses:
  IPv6 Unicast Addresses:
    Addr: 2001:1:1::1
    Addr: 2001:2:1::1
    Total number of IPv6 addresses:      2
  IPv6 Multicast Addresses:
    MAC: 3333FF280300  Addr: FF02::1:FF28:300
    Total number of IPv6 addresses:      1
Ancillary Input Queue Routing Variables:
  Queue Type: BULKDATA  Queue ID:  2  Protocol: TCP
    Src: 2004:1:11::1..200
    Dst: 2001:1:3::1..200
    Total number of IPv6 connections:      1
  Queue Type: SYSDIST   Queue ID:  3  Protocol: TCP
    Addr: 2001:1:3::1
    Total number of IPv6 addresses:      1
32 OF 32 Lines Displayed
End of report
```

Example of dual definition interface reply:

```
EZD0031I TCP/IP CS V2R1  TCPIP Name: TCPCS     15:14:15
Display OSAINFO results for IntfName: LNK29D
PortName: DEV29D    PortNum: 01  Datapath: 3902   RealAddr: 0002
PCHID: 0451         CHPID: 29    CHPID Type: OSD  OSA code level: 6760
Gen: OSA-E3         Active speed/mode: 1000 mb/sec full duplex
Media: Singlemode Fiber        Jumbo frames: Yes  Isolate: No
PhysicalMACAddr: 643B88F30000  LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 1   Ancillary queues in use: 0
Connection Mode: Layer 3       IPv4: Yes  IPv6: Yes
SAPSup: 00010293               SAPEna: 00010293
IPv4 attributes:
  VLAN ID: N/A             VMAC Active: No
  Defined Router: Non      Active Router: No
  AsstParmsEna: 00215C66   OutCkSumEna: 00000000  InCkSumEna: 00000000
IPv6 attributes:
  VLAN ID: N/A             VMAC Active: Yes
  VMAC Addr: 643B88F30001  VMAC Origin: Cfg       VMAC Router: All
  AsstParmsEna: 00215C66   OutCkSumEna: 00000000  InCkSumEna: 00000000
Registered Addresses:
  IPv4 Unicast Addresses:
    ARP: Yes  Addr: 10.10.10.10
    Total number of IPv4 addresses:      1
  IPv4 Multicast Addresses:
    MAC: 01005E000001  Addr: 224.0.0.1
    Total number of IPv4 addresses:      1
  IPv6 Unicast Addresses:
    Addr: 2001:1:1::1
    Addr: 2001:2:1::1
    Total number of IPv6 addresses:      2
  IPv6 Multicast Addresses:
    MAC: 3333FF280300  Addr: FF02::1:FF28:300
    Total number of IPv6 addresses:      1
30 OF 30 Lines Displayed
End of report
```

Example of Layer 2 reply:

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report

```
EZD0031I TCP/IP CS V2R1  TCPIP Name: TCPCS     15:14:15
Display OSAINFO results for IntfName: EZ6OSM01
PortName: IUTMP0CB  PortNum: 00  Datapath: 3902   RealAddr: 0002
PCHID: 0451         CHPID: 29    CHPID Type: OSM  OSA code level: 6760
Gen: OSA-E3         Active speed/mode: 1000 mb/sec full duplex
Media: Singlemode Fiber        Jumbo frames: Yes  Isolate: Yes
PhysicalMACAddr: 643B88F30000  LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 1   Ancillary queues in use: 0
Connection Mode: Layer 2
SAPSup: 00010293               SAPEna: 00010293
Layer 2 attributes:
  VLAN ID: N/A             VMAC Active: Yes
  VMAC Addr: 820001AA0E2A  VMAC Origin: OSA
Registered Addresses:
  Layer 2 Multicast MAC Addresses:
    MAC: 3333FF010003
    MAC: 3333FF010002
    MAC: 3333FF010001
    MAC: 3333FFAA0E2A
    MAC: 333300000001
    Total number of Layer 2 MAC addresses:      5
23 of 23 lines displayed
End of report
```

Reply field descriptions:

Interface

Interface name from the display command.

Base section:

The Base section of the report is displayed if the BASE modifier is specified or none of the modifiers are specified.

PortName

Portname specified on the INTERFACE definition, specified as the device name, or both when the datapath device is shared by both definitions. This name also matches *portname* on the VTAM TRLE definition.

PortNum

Physical port on the OSA-Express that is used for the interface.

Datapath

Hexadecimal datapath device address on the OSA-Express that is used for the interface.

RealAddr

Hexadecimal logical address and unit address of the interface.

PCHID

Physically installed channel path that is used by this QDIO datapath device.

CHPID

Channel path identifier that is used by this QDIO datapath device.

CHPID Type

The CHPID type of the interface, which can have the following values:

OSD External network

OSM Intra node management network

OSX Intra ensemble data network

OSA code level
OSA-Express processor code level of the QDIO datapath device.

Gen Generation of the OSA-Express feature. The following values are supported:

OSA-E3
OSA-Express3

OSA-E4S
OSA-Express4S

Active speed/mode
Switch speed and duplex mode of the interface. The following values are supported:

- 10 mb/sec half duplex
- 10 mb/sec full duplex
- 100 mb/sec half duplex
- 100 mb/sec full duplex
- 1000 mb/sec half duplex
- 1000 mb/sec full duplex
- 10 gigabit full duplex
- Unknown

Media Transmission media (copper or fiber). If fiber is the transmission media, it can be single-mode fiber (LR/LX) or multimode (SR/SX). The following values are supported:

- Copper
- Multimode Fiber
- Single-mode Fiber

Jumbo frames
Indicates whether jumbo frames are supported.

Isolate
Indicates whether this TCP/IP stack is prohibited from communicating directly through the interface with other TCP/IP stacks that are sharing the OSA-Express feature.

PhysicalMACAddr
Physical Medium Access Control (MAC) LAN address for the interface.

LocallyCfgMACAddr
Local Medium Access Control (MAC) LAN address for the interface.

Queues defined

Out Number of output priority queues that are defined for this interface.

In Number of input queues that are defined for this interface. A value of 1 indicates only the primary queue is defined. A value larger than 1 indicates that QDIO inbound workload queueing ancillary queues are defined and the number of ancillary queues is 1 less than the value reported.

Ancillary queues in use
Number of QDIO inbound workload queueing ancillary input queues (SYSDIST, BULKDATA, and so on) in use by this interface.

Connection Mode
Connection mode of the interface. The following values are supported:

- Layer 2
- Layer 3

IPv4 Indicates whether an IPv4 interface is active for the datapath device.

IPv6 Indicates whether an IPv6 interface is active for the datapath device.

SAPSup
Information used for problem analysis by IBM support.

SAPEna
Information used for problem analysis by IBM support.

IPv4, IPv6, or Layer 2 attributes
This section displays the attributes for the interface.

VLAN ID
Decimal virtual LAN identification number that is defined on this interface.

VMAC Active
Indicates whether the interface is using a virtual MAC address.

Defined Router
The defined router attribute. This field is displayed for Layer 3 only when VMAC is not active. The following values are supported:

Pri The interface is a primary router.

Sec The interface is a secondary router.

Non The interface is not a router.

Active Router
Indicates whether this interface is the active router for the OSA-Express feature. This field is displayed for Layer 3 only when VMAC is not active and is applicable only for PRIROUTER and SECROUTER interface configurations.

VMAC Addr
Displays the virtual MAC address in use for this interface. This field is displayed only when VMAC is active.

VMAC Origin
Indicates the origin of the virtual MAC address. This field is displayed only when VMAC is active. The following values are supported:

Cfg The virtual MAC address was configured in the TCP/IP stack PROFILE

OSA The virtual MAC address was assigned by the OSA-Express

VMAC Router
This field is displayed for Layer 3 only when VMAC is active. The following values are supported:

All Indicates that the OSA-Express is routing everything that was received on the virtual MAC address to the interface without regard for registered addresses.

Local Indicates that the OSA-Express is routing everything that is received on the virtual MAC address, and to a registered IP address, to the interface.

AsstParms
This field is displayed only for Layer 3. It is information used for problem analysis by IBM support.

OutCkSumEna
This field is displayed only for Layer 3. It is information used for problem analysis by IBM support.

InCkSumEna
This field is displayed only for Layer 3. It is information used for problem analysis by IBM support.

Registered Addresses
This is the registered addresses section of the report and is displayed if the REGADDRS modifier is specified or none of the modifiers are specified and only if there are registered addresses.

For Layer 3, there are four subsections that are included only if there are addresses to report:

- IPv4 Unicast Addresses
- IPv4 Multicast Addresses
- IPv6 Unicast Addresses
- IPv6 Multicast Addresses

For Layer 2, there is only one subsection which is included only if there are addresses to report:

- Layer 2 Multicast MAC Addresses

ARP Indicates whether the OSA-Express is providing address resolution for the corresponding registered IPv4 address.

Addr IPv4 or IPv6 address.

Total number of IPv4 addresses or Total number of IPv6 addresses
Shows the cumulative number of IPv4 or IPv6 addresses immediately preceding this message.

MAC The Medium Access Control (MAC) LAN address corresponding to the Layer 2 or registered multicast IP address.

Total number of Layer 2 MAC addresses
Shows the cumulative number of MAC addresses immediately preceding this message.

Ancillary Input Queue Routing Variables
The Ancillary Input Queue Routing Variables section of the report is displayed if any of the following modifiers were specified, or none were specified:

- BULKDATA
- SYSDIST

Queue Type
Displays the workload name for an ancillary queue. The following values are supported:

BULKDATA
Specifies that the input queue is used for streaming workloads.

SYSDIST
Specifies that the input queue is used for sysplex distributor workloads.

Queue ID
Ancillary queue number.

Protocol
TCP

Src Source address and port. This information is displayed only for the BULKDATA queue.

Dst Destination address and port. This information is displayed only for the BULKDATA queue.

Total number of IPv4 connections or Total number of IPv6 connections
Displays the cumulative number of BULKDATA IPv4 or IPv6 Src/Dst combinations immediately preceding this message.

DISPLAY TCPIP,,STOR

Use the DISPLAY TCPIP,*procname*,STOR command to display TCP/IP storage usage information. You can use this command to verify the load module service level.

To verify load module service level, ensure that the eyecatcher for the module matches the latest PTF service for the module. When you contact IBM Service, you can use this command to verify that you are running on the correct TCP/IP service level.

Format:

```
►►──Display ──TCPIP──,──┬────────────┬──,──STOR──┬──────────────────────────┬──────────►◄
                        └─procname───┘           └─,──MODule=──modname_name─┘
```

Parameters:

STOR
Requests storage information.

If no other option is specified, the command displays the current and maximum storage usage for the TCP/IP stack and any TCP/IP storage limits. The maximum storage usage is the highest amount of storage TCP/IP has used since it started. See “Example” on page 111 for an example output, and see message EZZ8453I in z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM) for a description of the output displayed.

MODULE
Displays the load module name that contains the module, module address and the first 48 bytes of storage.

This command displays modules within load modules EZBTIINI, EZBITCOM, EZBPFINI, EZBTLMST, EZBTLCMN, and EZBTLCLG. This command does not provide information for the FTP TCP/IP modules.

Load module
Storage Location

EZBTIINI
Common storage

EZBITCOM
Common storage

EZBPFINI
OMVS private storage

EZBTLMST
TCP/IP private storage

EZBTLCMN
TCP/IP private storage

EZBTLCLG
TCP/IP private storage

Example:
To display TCP/IP storage usage, issue the following command:

```
d tcpip,tcpip2,stor
EZZ8453I TCPIP STORAGE
EZZ8454I TCPIP2   STORAGE            CURRENT   MAXIMUM     LIMIT
EZD2018I 31-BIT
EZZ8455I          ECSA                45654K    56823K   204800K
EZZ8455I          PRIVATE            124634K   143743K   524288K
EZZ8455I          ECSA MODULES         8702K     8702K   NOLIMIT
EZD2018I 64-BIT
EZZ8455I          HVCOMMON                3M        3M   NOLIMIT
EZZ8455I          HVPRIVATE              50M       50M   NOLIMIT
EZZ8455I          TRACE HVCOMMON       2578M     2578M     2578M
EZZ8455I          SMC-R FIXEDMEMORY     12M       16M       40M
EZD2024I            SMC-R SEND MEMORY     4M        4M
EZD2024I            SMC-R RECV MEMORY     8M       12M
EZZ8459I DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY
```

Usage:

- If a module is built into multiple load modules, each occurrence is displayed.
- The storage display command is used to verify the load module service level of the TCP/IP stack. The command supports several, but not all, modules within the product.
- SMC-R memory information (messages EZZ8455I and EZD2024I) is included only when the Shared Memory Communications over Remote Direct Memory Access (SMC-R) function is or was enabled on this TCP/IP stack. The SMC-R function is enabled by using the SMCR parameter of the GLOBALCONFIG statement.

DISPLAY TCPIP,,SYSPLEX

Use the DISPLAY TCPIP,,SYSPLEX command from an operator console to request SYSPLEX information.

Format:

```
►►──Display ──TCPIP──,──┬──────────┬──,──────────────────────────────────────►
                        └─procname─┘
```

```
                                                    ┌─,MAX=100──┐ (1)
►─SYSplex──,──┬─VIPADyn──┬─────────────────────┬────┼───────────┼──────┬──────────►◄
              │          ├─,IPAddr=──ipaddr────┤    ├─,MAX=*────┤      │
              │          └─,INTFName=──intfname┘    └─,MAX=recs─┘      │
              ├─GROUP──────────────────────────────────────────────────┤
              └─PORTS──────────────────────────────────────────────────┘
```

Notes:

1 MAX limits the number of records displayed to the MVS operator's console.

Result: If the stack is not a member of a sysplex group, the following message is displayed:

```
EZZ8269I tcpstackname mvsname IS NOT A MEMBER OF A SYSPLEX GROUP
```

Parameters:

SYSPLEX
Request SYSPLEX information.

VIPADYN
Displays information about Dynamic VIPA for the active stack. If more than one stack is active, use *procname* to specify the particular TCP stack for which you want to display information.

The display contains a Distribute field. This field indicates whether the stack is a distributing stack, a destination stack, or both. For a description of all the fields displayed on the report, see the output examples for the "Netstat VIPADyn/-v report" on page 645.

IPADDR=*ipaddr*
Specifies a fully qualified IPv4 or IPv6 address that is used to limit the VIPADYN option. No wildcard characters (* and ?) are allowed for this value.

INTFName=*intfname*
Specifies an IPv6 interface name that is used to limit the VIPADYN option.

MAX=*number of records*
Number of records to be written to the console. Valid values are in the range 1 - 65535. A wildcard (*) displays all records. The default value is 100.

GROUP
Displays the name of the TCP/IP sysplex group that the active stack has joined. If more than one stack is active, use the *procname* parameter to specify the particular TCP/IP stack for which you want to display information.

PORTS
Displays the configured EXPLICITBINDPORTRANGE port range (as specified on the GLOBALCONFIG EXPLICITBINDPORTRANGE statement) for this stack, and the currently active port range throughout the sysplex. If the stack is not configured for an explicit bind port range, a message is displayed to indicate that the range has not been configured on this stack. If this stack has not interrogated the active explicit bind port range, a message is displayed to indicate that the active range is not available from this stack.

Result: The range that was configured on this stack might not be the actual range that is in use throughout the sysplex at this time, because another stack that was started later with a different EXPLICITBINDPORTRANGE value configured (or a Vary Obey command specifying a file with a different EXPLICITBINDPORTRANGE value) can override the range that was configured by this stack.

Examples:
Not IPv6 enabled (SHORT format):

```
    d tcpip,tcpcs,sysplex,group

    EZZ8260I SYSPLEX CS V1R9
    EZZ8270I SYSPLEX GROUP FOR TCPCS   AT MVS004 IS EZBT1121
d tcpip,tcpcs,sysplex,ports

    EZD1293I Configured EXPLICITBINDPORTRANGE: 05000-06023
    EZD1294I Active EXPLICITBINDPORTRANGE: 07000-09047
    d tcpip,tcpcs,sysplex,vipadyn

    EZZ8260I SYSPLEX CS V1R9 513
    VIPA DYNAMIC DISPLAY FROM TCPCS    AT MVS004
    IPADDR: 201.2.10.11  LINKNAME: VIPLC9020A0B
      ORIGIN: VIPADEFINE
      TCPNAME  MVSNAME  STATUS RANK ADDRESS MASK    NETWORK PREFIX  DIST
      -------- -------- ------ ---- --------------- --------------- ----
      TCPCS    MVS004   ACTIVE      255.255.255.240 201.2.10.0      BOTH
      TCPCS2   MVS004   BACKUP 100                                  DEST
      TCPCS3   MVS005   BACKUP 010                                  DEST
   IPADDR: 201.2.10.12  LINKNAME: VIPLC9020A0C
      ORIGIN: VIPADEFINE
      TCPNAME  MVSNAME  STATUS RANK ADDRESS MASK    NETWORK PREFIX  DIST
      -------- -------- ------ ---- --------------- --------------- ----
      TCPCS    MVS004   ACTIVE      255.255.255.240 201.2.10.0      DIST
      TCPCS2   MVS004   ACTIVE                                      DEST
      TCPCS3   MVS005   BACKUP 010
   IPADDR: 201.2.10.13
      ORIGIN: VIPABACKUP
      TCPNAME  MVSNAME  STATUS RANK ADDRESS MASK    NETWORK PREFIX  DIST
      -------- -------- ------ ---- --------------- --------------- ----
      TCPCS2   MVS004   ACTIVE      255.255.255.192 201.2.10.0      DIST
      TCPCS    MVS004   MOVING                                      DEST
      TCPCS3   MVS005   BACKUP 010
   IPADDR: 201.2.10.21
      ORIGIN: VIPABACKUP
      TCPNAME  MVSNAME  STATUS RANK ADDRESS MASK    NETWORK PREFIX  DIST
      -------- -------- ------ ---- --------------- --------------- ----
      TCPCS3   MVS005   ACTIVE      255.255.255.192 201.2.10.0
      TCPCS2   MVS004   BACKUP 100
      TCPCS    MVS004   BACKUP 080
   IPADDR: 201.2.10.22
      ORIGIN: VIPABACKUP
      TCPNAME  MVSNAME  STATUS RANK ADDRESS MASK    NETWORK PREFIX  DIST
      -------- -------- ------ ---- --------------- --------------- ----
      TCPCS3   MVS005   ACTIVE      255.255.255.192 201.2.10.0
      TCPCS    MVS004   BACKUP 080
      TCPCS2   MVS004   QUIESC
   15 OF 15 RECORDS DISPLAYED
```

IPv6 enabled:

```
D TCPIP,TCPCS,SYSPLEX,VIPADYN
EZZ8260I SYSPLEX CS V1R9 711
VIPA DYNAMIC DISPLAY FROM TCPCS    AT MVS004
LINKNAME: VIPLC9020A0B
IPADDR/PREFIXLEN: 201.2.10.11/28
  ORIGIN: VIPADEFINE
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS    MVS004   ACTIVE      BOTH
  TCPCS2   MVS004   BACKUP 100  DEST
  TCPCS3   MVS005   BACKUP 010  DEST
LINKNAME: VIPLC9020A0C
IPADDR/PREFIXLEN: 201.2.10.12/28
  ORIGIN: VIPADEFINE
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS    MVS004   ACTIVE      DIST
  TCPCS2   MVS004   ACTIVE      DEST
  TCPCS3   MVS005   BACKUP 010
IPADDR: 201.2.10.13
  ORIGIN: VIPABACKUP
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS2   MVS004   ACTIVE      DIST
  TCPCS    MVS004   MOVING      DEST
  TCPCS3   MVS005   BACKUP 010
IPADDR: 201.2.10.21
  ORIGIN: VIPABACKUP
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS3   MVS005   ACTIVE
  TCPCS2   MVS004   BACKUP 100
  TCPCS    MVS004   BACKUP 080
IPADDR: 201.2.10.22
  ORIGIN: VIPABACKUP
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS3   MVS005   ACTIVE
  TCPCS    MVS004   BACKUP 080
  TCPCS2   MVS004   QUIESC
INTFNAME: DVIPA1
IPADDR: 2001:0DB8:1::1
  ORIGIN: VIPADEFINE
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS    MVS004   ACTIVE      BOTH
  TCPCS3   MVS005   ACTIVE      DEST
  TCPCS2   MVS004   ACTIVE      DEST
INTFNAME: DVIPA2
IPADDR: 2001:0DB8:2::2
  ORIGIN: VIPADEFINE
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS    MVS004   ACTIVE      BOTH
  TCPCS3   MVS005   ACTIVE      DEST
  TCPCS2   MVS004   ACTIVE      DEST
INTFNAME: DVIPA3
IPADDR: 2001:0DB8:3::3
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS2   MVS004   ACTIVE
INTFNAME: DVIPA4
IPADDR: 2001:0DB8:4::4
  TCPNAME  MVSNAME  STATUS RANK DIST
  -------- -------- ------ ---- ----
  TCPCS3   MVS005   ACTIVE
9 OF 9 RECORDS DISPLAYED
```

DISPLAY TCPIP,,TRACE

Use the DISPLAY TCPIP,,TRACE command to display information about network management applications that use the real-time application-controlled TCP/IP trace network management interface (NMI) to obtain real-time network management data from the TCP/IP stack. See Real-time application-controlled

TCP/IP trace NMI (EZBRCIFR) in z/OS Communications Server: IP Programmer's Guide and Reference for more information.

This command displays each application that currently uses the NMI, along with information about the storage that the TCP/IP stack uses to provide the real-time data. If neither the NAME parameter nor the TYPE parameter is specified, the command displays a statistics section that applies to all applications that currently use the NMI. Specify the DETAIL parameter of the command to display all the trace filters that have been specified by each application.

Format:

```
►►──Display ──TCPIP──,──┬──────────┬──,─────────────────────────────────►
                        └─procname─┘


          ┌────────────────────────┐
          ▼                        │ ┌─,MAX=200───┐
►─TRACE───┬──────────────────────┬─┴─┼────────────┼────────────────────►◄
          ├─,──DETAIL────────────┤   ├─,MAX=*─────┤
          ├─,──NAME=-jobname─────┤   └─,MAX=lines─┘
          └─,──TYPE=-──┬─DAT─┬───┘
                       └─PKT─┘
```

Rule: The parameters must be specified in the order that is shown in the syntax diagram.

Parameters:

DETAIL
Specifies that all the filter sets for each trace type will be displayed. If this parameter is not specified, none of the filter set information is displayed.

NAME=*jobname*
Specifies that only the information for the specified job name will be displayed. The *jobname* value can be one of the following MVS job names:

- The job name of the application that is using the real-time application-controlled TCP/IP trace NMI.
- The job name of the original application that created the application that is using the NMI. For example, if JOB1 forked JOB2 and JOB2 is using the NMI, you can specify either JOB1 or JOB2 as the job name. If fork or spawn services are not used, the JOB1 and JOB2 job names should be identical.

Guideline: If you specify the NAME parameter, only the information about the applications that are using the NMI is displayed. The NMI statistics section is not displayed.

TYPE=DAT|PKT
Specifies that only the information for jobs that use the NMI for the specified trace type will be displayed. If you specify both the DETAIL parameter and the TYPE parameter, all the filter sets for the specified trace type are displayed. The valid TYPE values currently supported are:

DAT Specifies that only the information for jobs using the NMI data trace function will be displayed.

PKT Specifies that only the information for jobs using the NMI packet trace function will be displayed.

Guideline: If you specify the TYPE parameter, only the information about the applications that are using the NMI is displayed. The NMI statistics section is not displayed.

MAX=*|*lines*

The maximum number of lines to be displayed on the console. Valid values are in the range 2 - 65533. Specify an asterisk (*) to display up to 65533 lines.

MAX=*

If MAX=* is specified and the report is truncated because the multi-line WTO maximum was exceeded, the following message is displayed:

```
Report truncated due to greater than 65533 lines of output
```

MAX=*lines*

The total number of lines that are displayed and the total number of lines that could have been displayed are shown in the following output line:

```
n of m lines displayed
```

where *n* is the number of lines that are displayed and *m* is the total number of lines that could have been displayed.

Examples:

1. Display the statistics section and summary information for all applications that are using the real-time application-controlled TCP/IP trace NMI.

```
D TCPIP,TCPCS,TRACE

EZD2016I DISPLAY TRACE for TCPCS
Real-time application-controlled TCP/IP trace NMI statistics
  TotStage:    0000000030M
  Collection buffer statistics
    TotalSize:  0000000064M          InUse:   0000000000K
    HiWater:    0000000318K          Lost:    00000000000000000000
Name: NMAPP2    OrigName: NMAPPL    ASID: 001A    UserID: JONES
  Description:  Network Management Collector      Active: Yes
  Mode:         Locate              Ecb:  Yes
  Staging buffer statistics
    TotalSize:  0000000020M          InUse:   0000000080K
    HiWater:    0000000014M          Lost:    00000000000000000000
    PutRecs:    00000000000000001287 GetRecs: 00000000000000001263
  Data trace:   Yes    FiltNum: 01   ATTLSClr: Yes
  Packet trace: Yes    FiltNum: 01   IPSecClr: No
Name: PKTMON    OrigName: PKTMON    ASID: 002A    UserID: SMITH
  Description:  Packet Monitor                    Active: No
  Mode:         Move                Ecb:  No
  Staging buffer statistics
    TotalSize:  0000000010M          InUse:   0000000005M
    HiWater:    0000000009M          Lost:    00000000000000000254
    PutRecs:    00000000000000002513 GetRecs: 00000000000000001965
  Data trace:   No     FiltNum: 00   ATTLSClr: No
  Packet trace: Yes    FiltNum: 01   IPSecClr: Yes
25 of 25 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

2. Display detailed information for all applications that are using the real-time application-controlled TCP/IP trace NMI to obtain packet trace information.

```
D TCPIP,TCPCS,TRACE,TYPE=PKT,DETAIL

EZD2016I DISPLAY TRACE for TCPCS
```

```
Name: NMAPP2    OrigName: NMAPPL    ASID: 001A    UserID: JONES
  Description:  Network Management Collector      Active: Yes
  Mode:         Locate              Ecb:  Yes
  Staging buffer statistics
    TotalSize:  0000000020M          InUse:   0000000080K
    HiWater:    0000000014M          Lost:    00000000000000000000
    PutRecs:    00000000000000001287 GetRecs: 00000000000000001263
  Data trace:   Yes    FiltNum: 01   ATTLSClr: No
  Packet trace: Yes    FiltNum: 02   IPSecClr: Yes
    Packet trace filter set #    1
      Protocol: TCP                  Payload: 0000200
      Discard:  None                 PortNum: Any
      IPSec:    None
      IntfName: OSAINTF1
      IpAddr:   Any                                      PfxLen: None
    Packet trace filter set #    2
      Protocol: Any                  Payload: 0065535
      Discard:  None                 PortNum: Any
      IPSec:    Clear
      IntfName: Any
      IpAddr:   10.143.7.65                              PfxLen: None
Name: PKTMON    OrigName: PKTMON    ASID: 002A    UserID: SMITH
  Description:  Packet Monitor                    Active: No
  Mode:         Move                Ecb:  No
  Staging buffer statistics
    TotalSize:  0000000010M          InUse:   0000000005M
    HiWater:    0000000009M          Lost:    00000000000000000254
    PutRecs:    00000000000000002513 GetRecs: 00000000000000001965
  Data trace:   No     FiltNum: 00   ATTLSClr: No
  Packet trace: Yes    FiltNum: 02   IPSecClr: No
    Packet trace filter set #    1
      Protocol: Any                  Payload: 0065535
      Discard:  None                 PortNum: Any
      IPSec:    Secure
      IntfName: Any
      IpAddr:   10.2.104.126                             PfxLen: None
    Packet trace filter set #    2
      Protocol: Any                  Payload: 0065535
      Discard:  None                 PortNum: Any
      IPSec:    Secure
      IntfName: Any
      IpAddr:   2001:0DB8:0:1:10:2:105:254               PfxLen: None
44 of 44 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

3. Display detailed information for all applications that are using the real-time application-controlled TCP/IP trace NMI to obtain data trace information.

```
D TCPIP,TCPCS,TRACE,TYPE=DAT,DETAIL
```

```
EZD2016I DISPLAY TRACE for TCPCS
Name: NMAPP2    OrigName: NMAPPL    ASID: 001A    UserID: JONES
  Description:  Network Management Collector      Active: Yes
  Mode:         Locate              Ecb:  Yes
  Staging buffer statistics
    TotalSize:  0000000020M          InUse:   0000000080K
    HiWater:    0000000014M          Lost:    00000000000000000000
    PutRecs:    00000000000000001287 GetRecs: 00000000000000001263
  Data trace:   Yes    FiltNum: 02   ATTLSClr: Yes
    Data trace filter set #    1
      JobName:  TSTAPPL              Payload: 0065535
      ATTLS:    None                 PortNum: Any
      IpAddr:   2001:0DB8:0:1:10:2:105:254               PfxLen: None
    Data trace filter set #    2
      JobName:  DBAPPL               Payload: 1000000
      ATTLS:    Clear                PortNum: 00163
      IpAddr:   10.5.126.0                               PfxLen: 0024
  Packet trace: Yes    FiltNum: 01   IPSecClr: No
19 of 19 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

4. Display summary information for a specific application.

```
D TCPIP,TCPCS,TRACE,NAME=NMAPP2
```

```
EZD2016I DISPLAY TRACE for TCPCS
Name: NMAPP2    OrigName: NMAPPL    ASID: 001A    UserID: JONES
  Description:  Network Management Collector      Active: Yes
  Mode:         Locate              Ecb:  Yes
  Staging buffer statistics
    TotalSize:  0000000020M          InUse:   0000000080K
    HiWater:    0000000014M          Lost:    00000000000000000000
    PutRecs:    00000000000000001287 GetRecs: 00000000000000001263
  Data trace:   Yes    FiltNum: 01   ATTLSClr: No
  Packet trace: Yes    FiltNum: 02   IPSecClr: Yes
11 of 11 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

5. Display detailed information for a specific application.

```
D TCPIP,TCPCS,TRACE,NAME=NMAPP2,DETAIL
```

```
EZD2016I DISPLAY TRACE for TCPCS
Name: NMAPP2    OrigName: NMAPPL    ASID: 001A    UserID: JONES
  Description:  Network Management Collector      Active: Yes
  Mode:         Locate              Ecb:  Yes
  Staging buffer statistics
    TotalSize:  0000000020M          InUse:   0000000080K
    HiWater:    0000000014M          Lost:    00000000000000000000
    PutRecs:    00000000000000001287 GetRecs: 00000000000000001263
  Data trace:   Yes    FiltNum: 02   ATTLSClr: Yes
    Data trace filter set #    1
      JobName:  TSTAPPL              Payload: 0065535
      ATTLS:    None                 PortNum: Any
      IpAddr:   2001:0DB8:0:1:10:2:105:254              PfxLen: None
    Data trace filter set #    2
      JobName:  DBAPPL               Payload: 1000000
      ATTLS:    Clear                PortNum: 00163
      IpAddr:   10.5.126.0                              PfxLen: 0024
  Packet trace: Yes    FiltNum: 02   IPSecClr: Yes
    Packet trace filter set #    1
      Protocol: TCP                  Payload: 0000200
      Discard:  None                 PortNum: Any
      IPSec:    None
      IntfName: OSAINTF1
      IpAddr:   Any                                     PfxLen: None
    Packet trace filter set #    2
      Protocol: Any                  Payload: 0065535
      Discard:  None                 PortNum: Any
      IPSec:    Clear
      IntfName: Any
      IpAddr:   10.143.7.65                             PfxLen: None
31 of 31 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

6. Display the information when the real-time application-controlled TCP/IP trace NMI is active, but has not been used by any application yet.

```
D TCPIP,TCPCS,TRACE
```

```
EZD2016I DISPLAY TRACE for TCPCS
Real-time application-controlled TCP/IP trace NMI available for use
3 of 3 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

7. Display the information when the real-time application-controlled TCP/IP trace NMI is disabled.

```
D TCPIP,TCPCS,TRACE

EZD2016I DISPLAY TRACE for TCPCS
Real-time application-controlled TCP/IP trace NMI disabled
3 of 3 lines displayed
EZZ0053I COMMAND DISPLAY TCPIP,,TRACE COMPLETED SUCCESSFULLY
```

Report field descriptions:

- NMI Global information

  TotStage
  The total amount of trace instance staging buffer storage that is currently allocated. Each trace instance specifies the amount of storage to be allocated on the RCCOpen request.

  Collection buffer statistics section

  TotalSize
  The size of the collection buffer in megabytes.

  InUse The number of kilobytes or megabytes currently in use in the collection buffer.

  HiWater
  The largest number of kilobytes, or megabytes that have been in use in the collection buffer since an application opened a trace instance.

  Lost The total number of trace records that could not be written to the collection buffer due to insufficient storage space.

- Application information section

  Name The job name of the application that opened an NMI trace instance. Applications invoke the RCCOpen NMI request to open a trace instance. If an application opens several trace instances, the same job name is displayed for all the trace instances.

  OrigName
  The job name of the originator of the application. For example, if JOB1 forked JOB2 and JOB2 is using the NMI, the job name JOB1 will be displayed in this field. If fork or spawn services are not being used, the value of this field is the same as the value of the Name field.

  ASID The hexadecimal address space identifier of the application that opened an NMI trace instance.

  UserID
  The user ID that is associated with the application that opened an NMI trace instance.

  Description
  The description that was supplied by the application on the RCCOpen NMI request when it opened an NMI trace instance.

  Active Indicates whether the application trace instance is currently active. If the trace instance is active, trace records are being created in the staging buffer for the application to retrieve.

  Yes Indicates that the application trace instance is currently active.

No Indicates that the application trace instance is not currently active.

Mode Indicates the mode that the application trace instance uses to access the trace records.

Locate The application uses locate mode. This option is specified on the RCCOpen NMI request.

Move The application uses move mode. This is the default mode if locate mode is not specified on the RCCOpen NMI request.

Ecb Indicates whether the application trace instance uses an ECB that the application owns to be posted when trace records are written to the staging buffer.

Yes The application uses an ECB that the application owns. This option is specified on the RCCOpen NMI request.

No The application does not use an ECB that the application owns. This is the default option if an ECB address is not supplied on the RCCOpen NMI request.

Staging buffer statistics section

TotalSize
The size of the staging buffer in megabytes.

InUse The number of kilobytes or megabytes currently in use in the staging buffer. This value is reset to zero each time the RCCStart request is processed for the trace instance.

HiWater
The largest number of kilobytes, or megabytes that have been in use in the staging buffer since the last RCCStart request was processed for the trace instance.

Lost The total number of trace records and collection buffer lost records that could not be written to the staging buffer as the result of insufficient storage space. This value is reset to zero each time the RCCStart request is processed for the trace instance.

PutRecs
The total number of trace records that have been stored in the staging buffer since the last RCCStart request was processed for the trace instance.

GetRecs
The total number of trace records that the application has retrieved from the staging buffer since the last RCCStart request was processed for the trace instance.

Data trace filter control block section

Data trace
Indicates whether the application trace instance has set up filters for data trace.

Yes Indicates that the application trace instance has set up filters for data trace. If the DETAIL parameter was specified on the command, the filter sets will be displayed under this field.

No Indicates that the application trace instance has not set up filters for data trace.

FiltNum *num*
Indicates the total number of filters that are defined for the trace type.

ATTLSClr
Indicates whether AT-TLS cleartext data is requested in any of the data trace filters.

Yes Indicates that AT-TLS cleartext data is requested in at least one of the filters.

No Indicates that AT-TLS cleartext data is not requested in any of the filters.

Data trace filter set #*num*
Indicates that the filter set number *num* is displayed under this field. The filter set information is displayed only if filters have been set for the trace type and the DETAIL parameter was specified on the command.

Payload
The specified or default payload value from the RCDAPayload field in the RCCDat filter control block.

Portnum
The port number from the RCDAPortNum field in the RCCDat filter control block. The port number applies to the source or destination port. One of the following values is displayed:

Any Indicates that no port number is specified for this filter set.

portnum
The port number.

ATTLS
The requested AT-TLS filter support from the RCDAFiltFlags field in the RCCDat filter control block. One or more of the following values can be displayed:

None Indicates that no AT-TLS filter support is requested.

Clear Indicates that AT-TLS cleartext data support is requested. Trace records will be created for AT-TLS before the data is encapsulated or encrypted.

Restriction: The data must match other filter criteria for a trace record to be created for it.

JobName
The job name from the RCDAJobName field in the RCCDat filter control block. One of the following values is displayed:

Any Indicates that no job name is specified for this filter set.

jobname
The job name.

IpAddr
The job name from the RCDAJobName field in the RCCDat filter control block. One of the following values is displayed:

Any Indicates that no job name is specified for this filter set.

jobname
The job name.

PfxLen
The prefix length from the RCDAPrefix field in the RCCDat filter control block. This value is used together with the IP address value to create an IP address mask that will be used in comparing the IP address to the source or destination IP addresses in a packet. One of the following values is displayed:

None Indicates that no prefix length is specified for this filter set.

pfxlen The prefix length in number of bits.

Packet trace filter control block section

Packet trace
Indicates whether the application trace instance has set up filters for the packet trace.

Yes Indicates that the application trace instance has set up filters for the packet trace. If the DETAIL parameter was specified on the command, the filter sets are displayed under this field.

No Indicates that the application trace instance has not set up filters for packet trace.

FiltNum *num*
Indicates the total number of filters that are defined for the trace type.

IPSecClr
Indicates whether IPSec cleartext data is requested in any of the data trace filters.

Yes Indicates that IPSec cleartext data is requested in at least one of the filters.

No Indicates that IPSec cleartext data is not requested in any of the filters.

Packet trace filter set #*num*
Indicates that filter set number *num* is displayed under this field. The filter set information is displayed only if filters have been set for the trace type and the DETAIL parameter was specified on the command.

Protocol
The protocol number from the RCPKProto field in the RCCPkt filter control block. One of the following values will be displayed:

Any Indicates that no protocol was specified for this filter set.

protocol | **TCP** | **UDP** | **ICMP** | **ICMPv6**
The protocol number or the protocol name for the indicated well-known protocols.

Payload
The specified or default payload value from the RCPKPayload field in the RCCPkt filter control block.

Discard
The discarded packets value from the RCPKDiscard field in the RCCPkt filter control block. One of the following values will be displayed:

None Indicates that trace records are not created for discarded packets.

Both Indicates that trace records are created for both discarded packets and packets that were sent and delivered.

Only Indicates that trace records are created only for discarded packets.

reason_code
Indicates that trace records are created only for packets with the specified discard reason code.

Portnum
The port number from the RCPKPortNum field in the RCCPkt filter control block. The port number applies to the source or destination port. One of the following values will be displayed:

Any Indicates that no port number was specified for this filter set.

portnum
The port number.

IPSec The requested IPSec filter support from the RCPKFiltFlags field in the RCCPkt filter control block. One or more of the following values can be displayed:

None Indicates that no IPSec filter support is requested.

Clear Indicates that IPSec cleartext data support is requested. Trace records will be created for IPSec packets before the packet is encapsulated or after the packet is decapsulated.

Secure
Indicates that IPSec secure data support is requested. Trace records will be created for encapsulated IPSec packets.

Restriction: The packet must match other filter criteria for a trace record to be created for it.

IntfName
The interface name from the RCPKIntfName field in the RCCPkt filter control block. One of the following values will be displayed:

Any Indicates that no interface name was specified for this filter set.

intfname
The interface name.

IpAddr
The IPv4 or IPv6 address from the RCPKIpAddr field in the RCCPkt filter control block. The IP address applies to the source or destination address. If a packet contains GRE headers, the IP address also applies to the source or destination address in the GRE header. One of the following values will be displayed:

Any Indicates that no IP address was specified for this filter set.

ipaddr The IP address.

PfxLen
The prefix length from the RCPKPrefix field in the RCCPkt filter control block. The NMI uses the prefix length value and the IP address value to create an IP address mask. The IP address mask is used to compare the IP address to the source or destination IP addresses in a packet. One of the following values will be displayed:

None Indicates that no prefix length was specified for this filter set.

pfxlen The prefix length in number of bits.

Matches
The number of trace records that matched the values in this filter set. This value includes normal trace records and discarded trace records.

DISPLAY command: TN3270E Telnet server address space

Use the DISPLAY TCPIP,*tnproc*,<TELNET> command from an operator console to request TN3270E Telnet server (Telnet) information. You must specify the Telnet procedure name. Because all commands are directed to the Telnet address space, the keyword TELNET can be omitted.

```
►►──Display TCPIP──,tnproc──┬─────────┬──┬─,CLientID───┬──────────────────────►◄
                            └─,Telnet─┘  ├─,OBJect─────┤
                                         ├─,PROFile────┤
                                         ├─,CONNection─┤
                                         └─,INACTLUS───┘
```

The IPv6 address format is accepted wherever an IP address is specified. The result might be no matches, but the IPv6 address format is always accepted.

If the z/OS UNIX domain name is set to AF_INET6 for IPv6 or the FORMAT LONG configuration statement is specified, then tabular style displays that contain client identifier data use a second line to display data; otherwise, the data is displayed on a single line. To ensure uniformity in the displays, if the second line format is in effect, then any display that contains client identifier data uses the 2–line format even if the data would fit on a single line. The following tabular displays are affected:

- D TCPIP,*tnproc*,<TELNET>,CLientID
- D TCPIP,*tnproc*,<TELNET>,OBJect
- D TCPIP,*tnproc*,<TELNET>,CONNection

All commands that contain the PROFILE= parameter are considered to be part of the profile group because the commands categorize (and display) the information based on the profile in which it is contained. All of these commands search all profiles that match the PROFILE= search criteria. Once a match is found, the other parameters are used to determine what is displayed for the profile.

Profile, connection, and port-related displays contain a port description line that identifies the port for the preceding data.

Telnet Display commands provide the following summary or detailed information at all levels:

- Connection

  D TCPIP,*tnproc*<,Telnet>,CONNection (Summary | Detail)
- Profile
  - D TCPIP,*tnproc*<,Telnet>,PROFile (Summary | Detail)
  - D TCPIP,*tnproc*<,Telnet>,OBJect (Summary | Detail)
  - D TCPIP,*tnproc*<,Telnet>,CLient ID (Summary | Detail)
- Port
  - D TCPIP,*tnproc*<,Telnet>,PROFile (Summary)
- Server
  - D TCPIP,*tnproc*<,Telnet>,PROFile (Summary)
  - D TCPIP,*tnproc*<,Telnet>,INACTLUS

Telnet displays use multiple console support (MCS) display lines. In the examples, a C indicates a control line and an L indicates a label line. When MCS is being used, control and label lines do not scroll off the screen.

Tip: All parameters after the command can be in any order. All commands are directed to the Telnet address space, which makes the TELNET parameter redundant and optional.

When you specify a TN3270E Telnet server as the *tnproc* value on the command, you can display information about the TN3270E Telnet server or about functions that are associated with the server.

The functions listed in Table 7 support the DISPLAY TCPIP command when it is directed to a TN3270E Telnet server.

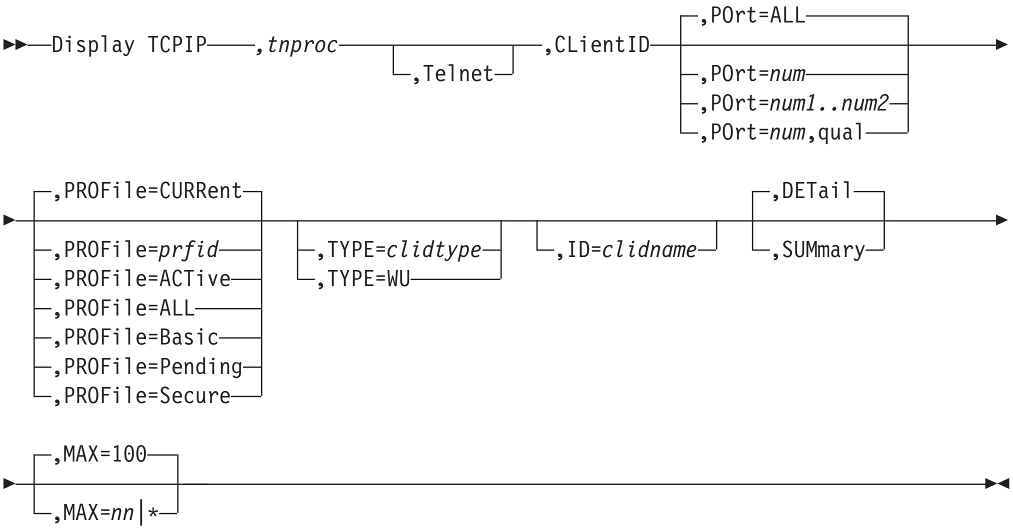
Table 7. TN3270E Telnet server functions that support the MVS DISPLAY TCPIP command

Function	**Command**
CLientID	“DISPLAY Telnet CLientID command” on page 126
CONNection	“DISPLAY TELNET CONNECTION command” on page 129
HELP	“DISPLAY TCPIP,*tnproc*,HELP” on page 133
INACTLUS	“DISPLAY Telnet INACTLUS command” on page 134
LUNS	“DISPLAY TCPIP,*tnproc*,LUNS” on page 135
OBJect	“DISPLAY Telnet OBJect command” on page 138
PROFile	“DISPLAY Telnet PROFILE command” on page 141
STOR	“DISPLAY TCPIP,*tnproc*,STOR” on page 144
Server status	“DISPLAY TCPIP,TELNET” on page 145
XCF	“DISPLAY TCPIP,*tnproc*,XCF” on page 146

DISPLAY Telnet CLientID command

Use the CLIENTID display command to display Client IDs that are defined in the profile and details about the Client ID.

Format:



Parameters:

tnproc
: The member name of the cataloged procedure that is used to start the Telnet address space.

Telnet
: Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

CLientID
: The CLientID keyword.

POrt=ALL|*num*|*num1..num2*|*num*,qual
: Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), qualified port (*num*,**qual**) be displayed. **ALL** is the default.

PROFile =CURRent|*prfid*|ACTive|ALL|Basic|Pending|Secure
: The type of profile to display.

- CURRent is the name of the current profile. This is the default.
- *prfid* is the profile ID.
- ACTive is all the active profiles.
- ALL is all profiles, both active and inactive.
- Secure is the secure profiles.
- Pending is the profile that is waiting for LUNS acknowledgement to become the active profile.
- Basic is the basic profile.

TYPE=*clidtype*
: The type of client identifier to display. The client identifier values are:

- USERID
- HOSTNAME
- IPADDR
- USERGRP
- HNGRP
- IPGRP
- DESTIP
- LINKNAME
- DESTIPGRP
- LINKGRP
- USERS (USERID and USERGRP)
- HNS (HOSTNAME and HNGRP)
- IPS (IPADDR and IPGRP)
- DESTIPS (DESTIP and DESTIPGRP)
- LINKS (LINKNAME and LINKGRP)
- NULL
- WU (Determines all the places where a particular name or IPADDR was used and presents mapping information.)

ID=*clidname*

The client identifier name. If more than one client ID has the same name, one line mapping information is displayed for all, but only the first one found in a random search will have details presented. Use TYPE with ID to get the correct match.

DETail|SUMmary

Summary is the default when neither TYPE nor ID is specified. Detail is the default if either TYPE or ID are specified. The following describes the different conditions:

- Neither TYPE nor ID is specified

  Summary

  Using message EZZ6082I, produces a listing of client identifiers.

  Detail

  Using message EZZ6081I, produces a more detailed display showing all Client Identifiers and the objects that are mapped to them.

- TYPE is specified

  Summary

  Using message EZZ6082I, produces a list of all Client Identifiers for the specified Client Identifier type.

  Detail

  Using message EZZ6081I, produces a more detailed display showing all Client Identifiers and the objects mapped to them for the specified client identifier type.

- ID is specified with or without TYPE

  Summary

  Using message EZZ6081I, produces a detailed display showing all client identifiers and the objects mapped to them for the specified client identifier.

Detail
: Using message EZZ6081I, produces a detailed display showing all client identifiers and the objects mapped to them for the specified client identifier. In addition, if the client identifier is a group, the individual client identifiers within the group are displayed. If a PARMSGROUP group is mapped to the client identifier, a summary of the resulting parameters used by a connection are displayed.

MAX=100|*nn*|*
: The number of output lines that are displayed. Valid values are in the range 2 - 65533. The default value is 100. An asterisk (*) means that all output lines are displayed. The command can display a maximum of 65533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65533 output lines is displayed.

Examples:

The following examples show what might be displayed with this command.

```
    D TCPIP,TELNET,CLIENTID,PORT=23,PROF=CURR,SUMMARY
(C) EZZ6082I TELNET CLIENTID LIST
    USERID
      USER10
    HOSTNAME
      TESTER12.RALEIGH.THIS.VERY.LONG.HOSTNAME.EXAMPLE.S
       HOWS.WRAP.COM
      TESTER11.ANYWHERE.IBM.COM
    IPADDR
      1.1.1.1
    HNGRP
      HNGRP1
    IPGRP
      IPGRP1
    LINKNAME
      CTCLNK6
    DESTIPGRP
      DIPGRP1
    ----- PORT:      23    ACTIVE           PROF: CURR CONNS:      0
    ------------------------------------------------------------
    18 OF 18 RECORDS DISPLAYED
```

```
D TCPIP,TELNET,CLIENTID,PORT=23,TYPE=HOSTNAME
(C) EZZ6081I TELNET CLIENTID DISPLAY
(L) CLIENT ID          CONNS  OBJECT    OBJECT   ITEM
(L) NAME               USING  TYPE      NAME     SPECIFIC   OPTIONS
    ----------------- ------ --------- -------- ---------- --------
    HOSTNAME
      TESTER12.RALEIGH.THIS.VERY.LONG.HOSTNAME.EXAMPLE.S
       HOWS.WRAP.COM
                            0 LU        LU12345             ----G---
                                                 TSO        D--L----
      TESTER11.ANYWHERE.IBM.COM
                            0 INT       EZBTPINT            --------
    ----- PORT:    23  ACTIVE               PROF: CURR CONNS:      0
    ------------------------------------------------------------
    10 OF 10 RECORDS DISPLAYED
```

```
    D TCPIP,TELNET,CLIENTID,PORT=23,ID=IPGRP1
    (C) EZZ6081I TELNET CLIENTID DISPLAY
    (L) CLIENT ID          CONNS  OBJECT    OBJECT   ITEM
    (L) NAME               USING  TYPE      NAME     SPECIFIC   OPTIONS
        ----------------- ------ --------- -------- ---------- --------
        IPGRP
          IPGRP1
                                0 DEFAPPL   APPL2               --------
          IPGRP1
```

```
                        0 LUGRP     LUGRP1              -C--G---
  IPGRP1
                        0 PRTGRP    PRTGRP1             ----GK--
  IPGRP1
                        0 PRT       PRT3333             ----GK--
  IPGRP1
                        0 PARMSGRP  PRMGRP2             --------
  IPGRP1
                        0 MONGRP    MONGRP1             --------
IPGRP: IPGRP1
     1.1.1.1
     2.2.2.2
     255.0.0.0:9.0.0.0
PARMS:
  PERSIS   FUNCTION      DIA  SECURITY   TIMERS  MISC
 (LMTGCAK)(OPATSKTQSSHRT)(DRF)(PCKLECXN2) (IPKPSTS)(SMLT)
  -------  ------------   ---  ---------  -- ----  ----
  *******  ***TSBTQ***RT   EC*  BB*******  *P**STS  *DD* *DEFAULT
  LM-----  -------------   DC-  ---------  -------  ---- *TGLOBAL
  ---R---  -P--------H--   ---  -B-------  -------  ---- *TPARMS
  -------  -------------   DJ-  ---------  -------  ---- PRMGRP2
  LM*R***  *P*TSBTQ**HRT   DJ*  BB*******  *P**STS  *DD* <-FINAL
----- PORT:    23  ACTIVE              PROF: CURR CONNS:      0
---------------------------------------------------------------
28 OF 28 RECORDS DISPLAYED
```

DISPLAY TELNET CONNECTION command

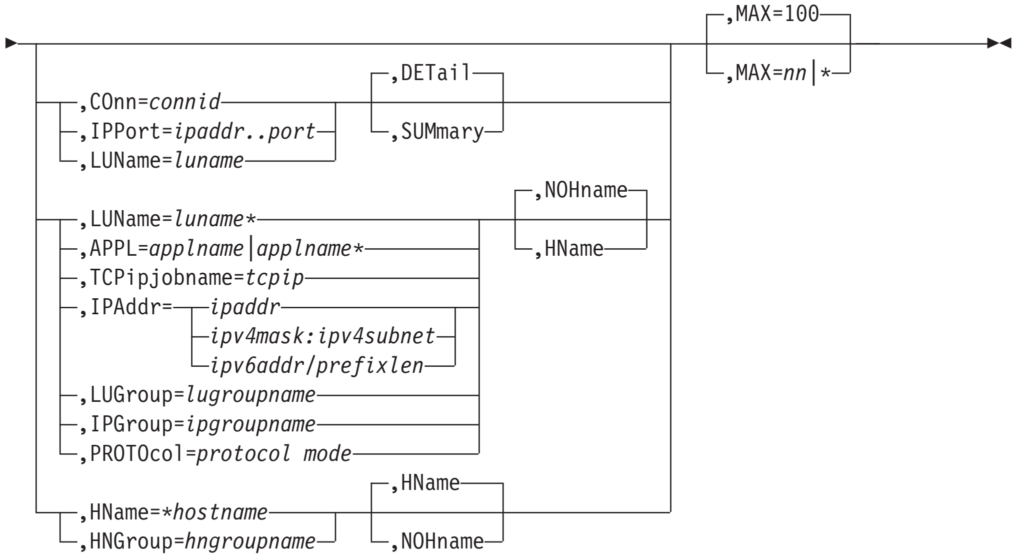
Use the CONNECTION DISPLAY command with the SUMmary parameter to view high-level information about multiple existing connections and their usage.

Use the CONNECTION DISPLAY command with the DETail parameter to view all available details about a single connection.

Use the LUName filter with the *LUNSREQ option to see the connections at a LUNR that are waiting for a reply from the LUNS.

Format:

```
►►── Display TCPIP───,tnproc────┬─────────┬─,CONNection───────────────────►
                                └─,Telnet─┘

   ┌─,POrt=ALL──────────┐  ┌─,PROFile=ALL──────┐
►──┼────────────────────┼──┼───────────────────┼──────────────────────────►
   ├─,POrt=num──────────┤  ├─,PROFile=prfid────┤
   ├─,POrt=num1..num2───┤  ├─,PROFile=ACTive───┤
   └─,POrt=num,qual─────┘  ├─,PROFile=CURRent──┤
                           ├─,PROFile=Basic────┤
                           └─,PROFile=Secure───┘
```



Parameters:

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

CONNection
The connection keyword.

POrt=ALL|*num*|*num1..num2*|*num*,qual
Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), qualified port (*num*,**qual**) be displayed. **ALL** is the default.

PROFile =ALL|*prfid*|ACTive|CURRent|Basic|Secure
The type of profile to display.

- ALL is all profiles, both active and inactive. This is the default.
- *prfid* is the profile ID.
- ACTive is all the active profiles.
- CURRent is the name of the current profile.
- Basic is the basic profile.
- Secure is the secure profiles.

COnn=*connid*
Displays detailed information about a specific TCP/IP connection ID.

IPPort=*ipaddr..port*
Displays detailed information about a specific IP port and address.

LUName=*luname**
The name of the LU for which you are searching. The wildcard (*) is allowed only as the last character of the LUName. If no * is indicated, a detailed display will appear.

SUMmary|DETail
DETail displays all of the information about the requested connection.

SUMmary displays a subset of the information about the requested connection.

APPL=*applname*|*applname**
The application name of the application for which you are searching. The wildcard (*) is allowed only as the last character.

TCPIPJOBNAME=*tcpip*
The TCPIP stack that supports the connection.

IPAddr=*ipaddr*|*mask:subnet*
The IP address of the connection for which you are searching. The `mask:subnet` designation is essentially allowing an IP wildcard.

LUGroup=*lugroupname*
The name of the LU group for which you are searching.

IPGroup=*ipgroupname*
The name of the IP group for which you are searching.

PROTOcol=*protocol mode*
The protocol mode for which you are searching. Protocol choices are:

- BINARY
- LINEMODE
- TN3270
- TN3270E
- TRANSFORM

HName|NOHname
The summary display includes client host names when HNAME is specified. The summary display omits client host names when NOHNAME is specified.

HName=*hostname*
The host name for which you are searching. Single or double asterisks are permitted as wildcards:

- Use a single asterisk (*) to indicate that any value is acceptable for a particular qualifier in a particular position within the host name. For example, *.*.IBM.COM matches USER1.RALEIGH.IBM.COM, but does not match USER1.TCP.RALEIGH.IBM.COM because this name includes an extra qualifier.
- Use a double asterisk (**) to indicate that any number of qualifiers are acceptable to the left of the asterisks. For example, **.IBM.COM matches USER1.IBM.COM, USER1.RALEIGH.IBM.COM, and USER1.TCP.RALEIGH.IBM.COM.

Both wildcard techniques require that the entire qualifier be wildcarded. For example, *USER.IBM.COM is not a valid use of a wildcard. In this case, use *.IBM.COM instead.

HNGroup=*hngroupname*
The name of the HN group for which you are searching.

MAX=100|*nn*|*****
The number of output lines that are displayed. Valid values are in the range 2–65533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65533 output lines is displayed.

Examples:

```
    D TCPIP,TELNET,CONN
(C) EZZ6064I TELNET CONN DISPLAY
(L)          ENCR                                           TSP
(L) CONN     TYPE IPADDR..PORT            LUNAME   APPLID   PTR LOGMODE
    -------- ---- ---------------------- -------- --------  --- --------
    0000001C 4S   9.27.11.197..4155      TCPM1002 APPL2     TAE SNX32702
    0000001A 4S   9.27.11.197..4154      TCPM1001 APPL2     TAE SNX32702
    ----- PORT:      23    ACTIVE          PROF: CURR CONNS:      2
    ------------------------------------------------------------
    4 OF 4 RECORDS DISPLAYED
```

The following example shows information that is related to AT-TLS.

```
D TCPIP,TELNET,CONN,CONN=35
EZZ6065I TELNET CONN DISPLAY
  CONNECTED: 12:01:49  10/26/2005  STATUS: SESSION ACTIVE
  CLIENT IDENTIFIER FOR CONN: 00000035   SECLABEL: **N/A**
    CLIENTAUTH USERID: USER60
    HOSTNAME: TEST3.IBM.COM
    CLNTIP..PORT: ::FFFF:9.16.17.18..2763
    DESTIP..PORT: ::FFFF:9.42.43.44..23
    LINKNAME: CTCLNK6
  PORT:    23 QUAL: NONE
    AFFINITY: TCPIP
    STATUS: ACTIVE  TTLSSECURE    ACCESS: SECURE  0005 SSLV3 SAFCHECK
    TTLSRule:       TTLSTNRULE1
    TTLSGrpAction:  TTLSTN3270GROUPACTION1
    TTLSEnvAction:  TTLSTN3270ENVIRONMENTACTION1
    TTLSConnAction: TTLSTN3270CONNECTIONACTION1
  PROTOCOL: TN3270E  LOGMODE: SNX32702 DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET----   3270E FUNCTIONS: BSR----
                      NEWENV FUNCTIONS: --
  USERIDS  RESTRICTAPPL: USER64    EXPRESSLOGON: **N/A**
  LUNAME: TCPM1011  TYPE: TERMINAL GENERIC  APPL: TSO10003
  MAPPING TYPE:  CONN IDENTIFIER
                      OBJECT   ITEM SPECIFIC     OPTIONS
    LUMAP GEN:  IG IPGRP1
                     >LUGRP1                     -E--G---
                      LUGRP2                     ----G---
    DEFLT APPL: IP ::FFFF:9.16.17.18
                      TSO                        --------
    USS TABLE:  IG IPGRP1
                      EZBTPUST                   P-------
                LU EXIT
                      EZBTPUST,>EZBTPSCS         PE------
    INT TABLE:  **N/A**
    MONGROUP:   IG IPGRP1
                      MONGRP1
      PERIOD:       60 MULT:        5
            S/W AVG TOT AVG  SUM R/T      SSQ R/T  ST DEV
            ======= ======= ======== ============ =======
      SNA:     2124    1316    17112     72757524    2046
      IP:         0       0        0            0       0
      TOTAL:   2124    1316    17112     72757524    2046
      COUNT:      4      13
      BUCKET1    BUCKET2    BUCKET3    BUCKET4    BUCKET5
           50        100        200        500     NO LMT
            1          1          0          1         10
    PARMS:
  PERSIS    FUNCTION        DIA   SECURITY    TIMERS    MISC
 (LMTGCAK) (OPATSKTQSSHRT) (DRF) (PCKLECXN2) (IPKPSTS) (SMLT)
  -------   ------------    ---   ---------   -------   ----
  *******   ***TSBTQ***RT   EC*   BB**D****   *P**STS   *DD* *DEFAULT
  -------   -------------   DJ-   -------*-   -------   S--- *TGLOBAL
 LM-R-P-   -----BT--WH--    ---   TSS--F---   ----ST-   ---- *TPARMS
 LM*R*P*   ***TSBTQ*-HRT    DJ*   TSS*DF***   *P**STS   SDD* TP-CURR
    PARMSGROUP: IG IPGRP1
  -------  OP-----------    TC-   --------2   I------   ---- PRMGRP1
    LUMAP-PMAP: ON LUMAP  OF LUGRP1
  -------  ------------     --F   ---------   -------   -M-- PRMGRP2
  LM*R*P*  OP*TSBTQ*-HRT    TCF   TSS*DF**2   IP**STS   SMD* <-FINAL
56 OF 56 RECORDS DISPLAYED
```

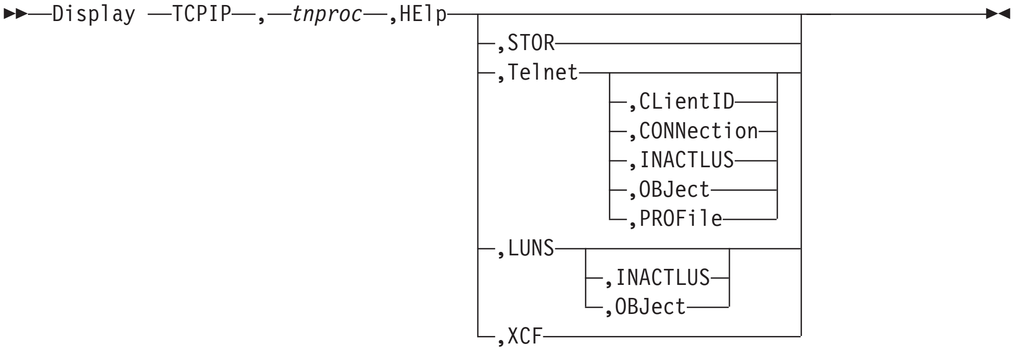
Usage:

Only one connection at a time is displayed with parameters CONN=, IPPort=, and LUName= if no wildcard is on LUName.

DISPLAY TCPIP,*tnproc*,HELP

Use the DISPLAY TCPIP,*tnproc*,HElp command from the MVS operator console to display the syntax of MVS operator DISPLAY commands for the TN3270E Telnet server (Telnet).

Format:



Parameters:

STOR
Shows help on the Telnet variation of the Display STOR command.

Telnet
Shows the available options on the Display Telnet command.

CLientID
Shows help on the Display TELNET,CLientID command.

CONNection
Shows help on the Display TELNET,CONNection command.

INACTLUS
Shows help on the Display TELNET,INACTLUS command or the Display LUNS,INACTLUS command.

OBJect
Shows help on the Display TELNET,OBJect command or the Display LUNS,OBJect command.

PROFile
Shows help on the Display TELNET,PROFile command.

LUNS
Shows help on the Display LUNS command.

XCF
Shows help on the Display XCF command.

Examples:

To view the available help for Telnet, issue the following information:

```
d tcpip,TNSERV,help,Telnet
EZZ6103I D TCPIP,TNPROC<,TELNET>,
(CLIENTID|CONNECTION|INACTLUS|OBJECT|PROFILE)
```

To get more information about the syntax of a particular Telnet command (for example, COnn), issue the following information:

```
d tcpip,TNSERV,help,telnet,conn
EZZ6107I D TCPIP,TNSERV<,TELNET>,CONNECTION
(<,(CONN=XCONNID|IPPORT=XIPADDR..XPORT|LUNAME=XLUNM)
  <,(DETAIL|SUMMARY)>>|
 <,(LUNAME=XLUNM*|APPL=(XAPPLNM|XAPPLNM*)|
    TCPIPJOBNAME=XTCPIPNM|PROTOCOL=XPROTMODE|
    LUGROUP=XLUGRPNM|IPGROUP=XIPGRPNM|
    IPADDR=(XIPADDR|XV4MASK:XV4SUBNET|XIPADDR/XPREFIXLEN))
  <,(NOHNAME|HNAME)>>|
 <,(HNAME=X*HOSTNAME|HNGROUP=XHNGROUPNM)
  <,(NOHNAME|HNAME)>>)
<,PORT=(ALL|XNUM|XNUM1..XNUM2|XNUM,XQUAL)>
<,PROF=(CURRENT|XPROFID|ACTIVE|ALL|BASIC|SECURE)>
<,SUMMARY|DETAIL>
<,MAX=(XNN|*)>
```

DISPLAY Telnet INACTLUS command

Use the INACTLUS DISPLAY command to see all of the LUs that have not been available to any users since the VARY INACT command was issued or since the OPEN ACB command failed and Telnet automatically set the LU state to inactive.

Format:

```
                                                     ┌─,MAX=100──┐
►►──Display TCPIP──,tnproc──┬──────────┬──,INACTLUS──┼───────────┼──────────────►◄
                            └─,Telnet──┘             └─,MAX=nn|*─┘
```

Parameters:

tnproc
   The member name of the cataloged procedure that is used to start the Telnet address space.

INACTLUS
   The inactive LUs keyword.

MAX=100|*nn*|*
   The number of output lines that are displayed. Valid values are in the range 2 - 65,533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65,533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65,533 output lines is displayed.

Examples:

```
D TCPIP,TELNET,INACTLUS
(C) EZZ6061I TELNET INACTLUS DISPLAY 771
(L) INACTIVE LUS
                TCPM1003  TCPM1005  TCPM1004  TCPM1001  TCPM1010
                TCPM1015  TCPM1012  TCPM1008
    5 OF 5 RECORDS DISPLAYED
```

DISPLAY TCPIP,*tnproc*,LUNS

Use the DISPLAY TCPIP,*tnproc*,LUNS command from an operator console to request TN3270E Telnet LU name server (LUNS) information. You must specify the Telnet procedure name.

Format:

```
                                                  ┌─,MAX=100──┐
►►──DISPLAY TCPIP──,tnproc──,LUNS──┬─,OBJect───┬──┼───────────┼──────────────────►◄
                                   └─,INACTLUS─┘  └─,MAX=nn|*─┘
```

The following descriptions provide details of the DISPLAY TCPIP,*tnproc*,LUNS commands that you can issue.

Display TCPIP,*tnproc*,LUNS,OBJect command:

Use the Display TCPIP,*tnproc*,LUNS,OBJect command to display the shared LU group objects at the LU name server (LUNS). The shared LU group objects are defined in the LU name requester (LUNR) profile and sent to the LUNS.

Format:

```
                                                   ┌─,POrt=ALL───────────┐
►►──Display TCPIP────,tnproc────,LUNS──,OBJect─────┼─────────────────────┼──────────────►
                                                   ├─,POrt=num───────────┤
                                                   ├─,POrt=num1..num2────┤
                                                   └─,POrt=num,qual──────┘

   ┌─,PROFile=CURRent─┐  ┌─,JOBname=ALL─────┐  ┌─,SYSname=ALL─────┐
►──┼──────────────────┼──┼──────────────────┼──┼──────────────────┼──────────────────►
   ├─,PROFile=prfid───┤  └─,JOBname=jobname─┘  └─,SYSname=sysname─┘
   └─,PROFile=ALL─────┘

                                          ┌─DETail──┐    ┌─,MAX=100──┐
►──┬────────────────┬──┬───────────────┬──┼─────────┼────┼───────────┼────────────►◄
   ├─,TYPE=objtype──┤  └─,ID=objname───┘  └─SUMmary─┘    └─,MAX=nn|*─┘
   └─,TYPE=WU───────┘
```

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

LUNS
The LUNS keyword.

OBJect
The OBJect keyword.

POrt=ALL|*num*|*num1*..*num2*|*num*,**qual**
Specifies that all ports, a specific port (*num*), a port number range (*num1*..*num2*), or a qualified port (*num*,qual) are to be displayed. The value ALL is the default.

PROFile = CURRent|*prfid*|**ALL**
The type of profile to display.

- CURRent is the name of the most recent profile that was received from the LUNR. This is the default.

- *prfid* is the profile ID.
- ALL indicates active profiles.

TYPE=*objtype*

The type of object identifier to display. Possible values are:

- SLUGRP
- SPRTGRP
- LUS (SLUGRP and SPRTGRP)
- WU (determines all places where a particular name is defined, presents mapping information, and displays where the name has been assigned.)

ID=*objname*

The shared object name. If more than one shared object has the same name, the first object found in a random search is presented. Specify the TYPE parameter with the ID parameter to get the correct match.

DETail|SUMmary

Summary is the default when neither the TYPE nor the ID parameter is specified. Detail is the default if either the TYPE or the ID parameter is specified. Possible conditions are:

- Neither TYPE nor ID is specified.

  Summary

  Using message EZZ6086I, produces a list of shared objects.

  Detail

  Using message EZZ6085I, produces a more detailed display that shows all shared objects.

- TYPE is specified.

  Summary

  Using message EZZ6086I, produces a list of all objects for the specified object type. Types SLUGRP and SPRTGRP provide a summary of total LUs and in-use LUs by group. An LU is considered in-use if it is assigned to a connection, is being kept for possible reuse, or is inactive. When TYPE=WU is specified, one line that shows where the LU is being used is displayed.

  Detail

  Using message EZZ6085I, produces a more detailed display showing all shared objects for the specified object type.

- ID is specified with or without the TYPE parameter.

  Summary

  Using message EZZ6085I, produces a detailed display showing all objects for the specified *objname* object name.

  Detail

  Using message EZZ6085I, produces a detailed display showing all objects for the specified object. If the object is a group, the individual objects in the group are displayed.

  MAX=100|*nn***|***

  The number of output lines that are displayed. Valid values are in the range 2 – 65 533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65 533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65 533 output lines is displayed.

Examples:

The following examples display possible output from this command.

```
   D TCPIP,TNLUNS,LUNS,OBJECT,PORT=6001,PROF=ALL
(C)EZZ6086I TNLUNS LUNS OBJECT LIST
   SLUGRP
     LUGRP1    LUGRP2    LUGRP3    LUGRP5    LUGRP6
   SPRTGRP
     *DEFPRT*  PRTGRP2   PRTGRP1
   ----------------------------------- PROF: 0001 CONNS:     23
   SLUGRP
     LUGRP1    LUGRP5    EZBLUXIT
   SPRTGRP
     *DEFPRT*  PRTGRP2   PRTGRP1
   ----- PORT:  6001 MVS024   TNLUNR1  PROF: 0002 CONNS:     16
   SLUGRP
     LUGRP1    LUGRP2    LUGRP6    LUGRP7    LUGRP9
   SPRTGRP
     *DEFPRT*  PRTGRP2   PRTGRP1
   ----- PORT:  6001 MVS024   TNLUNR2  PROF: 0002 CONNS:     11
   ------------------------------------------------------------
   18 OF 18 RECORDS DISPLAYED

   D TCPIP,TNLUNS,LUNS,OBJECT,TYPE=SLUGRP,SUMMARY,SYS=MVS024,JOB=TNLUNR2,PORT=6001
(C)EZZ6086I TNLUNS LUNS OBJECT LIST
   SLUGRP
      LUGRP1                                   20 LUS      7 IN USE
      LUGRP2                                  100 LUS     32 IN USE
      LUGRP6                                   10 LUS      2 IN USE
      LUGRP7                                   10 LUS      2 IN USE
      LUGRP9                                   10 LUS      2 IN USE
   ----- PORT:  6001 MVS024   TNLUNR2  PROF: 0002 CONNS:     11
   ------------------------------------------------------------
   10 OF 10 RECORDS DISPLAYED

   D TCPIP,TNLUNS,LUNS,OBJECT,TYPE=WU,ID=TCPM1008
(C)EZZ6085I TNLUNS LUNS OBJECT DISPLAY
(L)OBJECT      CONNS
(L)NAME        USING  OPTIONS
   ----------  ------ --------
   SLUGRP
    LUGRP1          2 --------
    LUGRP6          0 -C------
   ----------------------------------- PROF: 0001 CONNS:     23
   SLUGRP
    LUGRP1          0 --------
    LUGRP5          0 --------
   ----- PORT:  6001 MVS024   TNLUNR1  PROF: 0002 CONNS:     16
   SLUGRP
    LUGRP1          0 --------
    LUGRP6          1 --------
   SPRTGRP
    PRTGRP1         0 --------
   ----- PORT:  6001 MVS024   TNLUNR2  PROF: 0002 CONNS:     11
   LU: TCPM1008     STATUS: IN USE BY MVS024 TNLUNR1
   ------------------------------------------------------------
   21 OF 21 RECORDS DISPLAYED
```

Display TCPIP,*tnproc*,LUNS,INACTLUS command:

Use the D TCPIP,*tnproc*,LUNS,INACTLUS display command to see all of the LUs that have not been available to any LU name requesters because the VARY INACT command was issued or because the OPEN ACB failed and Telnet automatically set the LU state to inactive

Format:

```
                                        ┌─,MAX=100──┐
►►──Display TCPIP──,tnproc──,LUNS,INACTLUS──┼───────────┼──────────────────────────►◄
                                        └─,MAX=nn|*─┘
```

Parameters:

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

INACTLUS
The inactive LUs keyword.

MAX=100|*nn*|*
The number of output lines that are displayed. Valid values are in the range 2 – 65 533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65 533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65 533 output lines is displayed.

Examples:

```
D TCPIP,TNLUNS,LUNS,INACTLUS
(C) EZZ6062I TNLUNS LUNS INACTLUS
(L) INACTIVE LUS
               TCPM1003  TCPM1005  TCPM1004  TCPM1001  TCPM1010
               TCPM1015  TCPM1012  TCPM1008
    5 OF 5 RECORDS DISPLAYED
```

DISPLAY Telnet OBJect command

Use the OBJECT DISPLAY command to display objects that are defined in the profile and details about the object.

Format:

```
►►──Display TCPIP────,tnproc──┬─────────┬──,OBJect──────────────────────────►
                              └─,Telnet─┘

  ┌─,POrt=ALL─────────┐   ┌─,PROFile=CURRent─┐
►─┼───────────────────┼───┼──────────────────┼──┬──────────────────┬───────►
  ├─,POrt=num─────────┤   ├─,PROFile=prfid───┤  ├─,TYPE=objtype────┤
  ├─,POrt=num1..num2──┤   ├─,PROFile=ACTive──┤  └─,TYPE=WU─────────┘
  └─,POrt=num,qual────┘   ├─,PROFile=ALL─────┤
                          ├─,PROFile=Basic───┤
                          ├─,PROFile=Pending─┤
                          └─,PROFile=Secure──┘

                    ┌─,DETail───┐  ┌─,MAX=100──┐
►─┬───────────────┬─┼───────────┼──┼───────────┼─────────────────────────────►◄
  └─,ID=objname───┘ └─,SUMmary──┘  └─,MAX=nn|*─┘
```

Parameters:

tnproc
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

OBJect
The OBJect keyword.

POrt=ALL|*num*|*num1..num2*|*num*,qual
Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), qualified port (*num*,**qual**) be displayed. **ALL** is the default.

PROFile =CURRent|*prfid*|ACTive|ALL|Basic|Pending|Secure
The type of profile to display.

- CURRent is the name of the current profile. This is the default.
- *prfid* is the profile ID.
- ACTive is all the active profiles.
- ALL is all profiles, both active and inactive.
- Secure is the secure profiles.
- Pending is the profile waiting for LUNS acknowledgement to become the active profile.
- Basic is the basic profile.

TYPE=*objtype*
The type of object identifier to display. The object identifier values are:

- ARAPPL
- DEFAPPL
- INT
- LINEAPPL
- LU
- LUGRP
- SLUGRP
- MAPAPPL
- MONGRP
- PARMSGRP
- PRT
- PRTAPPL
- PRTGRP
- SPRTGRP
- USS
- APPLS (ARAPPL, DEFAPPL, PRTAPPL, LINEAPPL, MAPAPPL)
- DEFAULTS (DEFAPPL, PRTAPPL, LINEAPPL, MAPAPPL, USS, INT)
- LUS (LU, LUGRP, SLUGRP, APPLLUG, PRT, PRTGRP, SPRTGRP)
- WU (Determines all of the places where a particular name was used and presents mapping information.)

ID=*objname*
The object name. If more than one object has the same name, the first one found in a random search is presented. Use TYPE with ID to get the correct match.

DETail|SUMmary
Summary is the default when neither TYPE nor ID is specified. Detail is the default if either TYPE or ID are specified. The following describes the different conditions:

- Neither TYPE nor ID is specified

  Summary
  Using message EZZ6084I, produces a list of objects.

  Detail
  Using message EZZ6083I, produces a more detailed display showing all objects and the client identifiers to which they are mapped.

- TYPE is specified

  Summary
  Using message EZZ6084I, produces a list of all objects for the specified object type. Types LUGRP, PRTGRP, SLUGRP, and SPRTGRP provide a summary of total LUs and in-use LUs by group. An LU is considered to be in-use if it is assigned to a connection, is being kept for possible reuse, or is deactivated.

  Detail
  Using message EZZ6083I, produces a more detailed display showing all objects and the Client Identifiers to which they are mapped for the specified object type.

- ID is specified with or without TYPE

  Summary
  Using message EZZ6083I, produces a detailed display showing all objects and the Client Identifiers to which they are mapped for the specified object.

  Detail
  Using message EZZ6083I, produces a detailed display showing all objects and the Client Identifiers to which they are mapped for the specified object. In addition, if the object is a group, the individual objects within the group are displayed.

MAX=100|*nn***|***
The number of output lines that are displayed. Valid values are in the range 2 - 65 533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65 533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65 533 output lines is displayed.

Examples:

The following examples show what might be displayed with this command.

```
    D TCPIP,TELNET,OBJECT,PORT=23,SUMMARY
(C) EZZ6084I TELNET OBJECT LIST
    ARAPPL
      APPL1     APPL2     APPL3     APPL4
    DEFAPPL
      APPL1     APPL2
    MAPAPPL
```

```
      APPL2     TSO
    USS
      EZBTPUST
    INT
      EZBTPINT
    LU
      LU345     LU456     LU567     LU12345
    LUGRP
      *DEFLUS*  LUGRP1    LUGRP2
    PRT
      PRT12345  PRTGRP1   PRT3333
    PARMSGRP
      PRMGRP1   PRMGRP2   *DEFAULT  *TGLOBAL  *TPARMS
    ----- PORT:      23    ACTIVE          PROF: CURR CONNS:      0
    ------------------------------------------------------------
    20 OF 20 RECORDS DISPLAYED
D TCPIP,TELNET,OBJECT,PORT=23,TYPE=LUGRP
(C) EZZ6083I TELNET OBJECT DISPLAY
(L) OBJECT      CONNS  CLIENT ID CLIENT ID        ITEM
(L) NAME        USING  TYPE      NAME             SPECIFIC   OPTIONS
    ----------  ------ --------- ---------------- ---------- --------
    LUGRP
     *DEFLUS*        0
                                                             --------
     LUGRP1          0 IPGRP     IPGRP1
                                                             -C-LG---
     LUGRP1          0 LINKNAME  CTCLNK6
                                                             -C-LS---
                                                  APPL2      D---F---
     LUGRP2          0 HNGRP     HNGRP1
                                                             -C-LG---
    ----- PORT:    23  ACTIVE             PROF: CURR CONNS:      0
    ------------------------------------------------------------
    12 OF 12 RECORDS DISPLAYED
D TCPIP,TELNET,OBJECT,PORT=23,ID=LUGRP1
(C) EZZ6083I TELNET OBJECT DISPLAY
(L) OBJECT      CONNS  CLIENT ID CLIENT ID        ITEM
(L) NAME        USING  TYPE      NAME             SPECIFIC   OPTIONS
    ----------  ------ --------- ---------------- ---------- --------
    LUGRP
     LUGRP1          0 IPGRP     IPGRP1
                                                             -C-LG---
     LUGRP1          0 LINKNAME  CTCLNK6
                                                             -C-LS---
                                                  APPL2      D---F---
    LUGRP: LUGRP1   ,80%
    LU STATUS                            25354 LUS TOTAL
       TCPM1001  TCPM1002  TCPM1003
                                             3 LUS      0 IN USE
        TCPM1001..TCPM1008..FFFFFFFN         8 LUS      0 IN USE
        T01DPT01..T99DPTFF..FNNFFFXX     25343 LUS      0 IN USE
    ----- PORT:    23  ACTIVE             PROF: CURR CONNS:      0
    ------------------------------------------------------------
    12 OF 12 RECORDS DISPLAYED
```

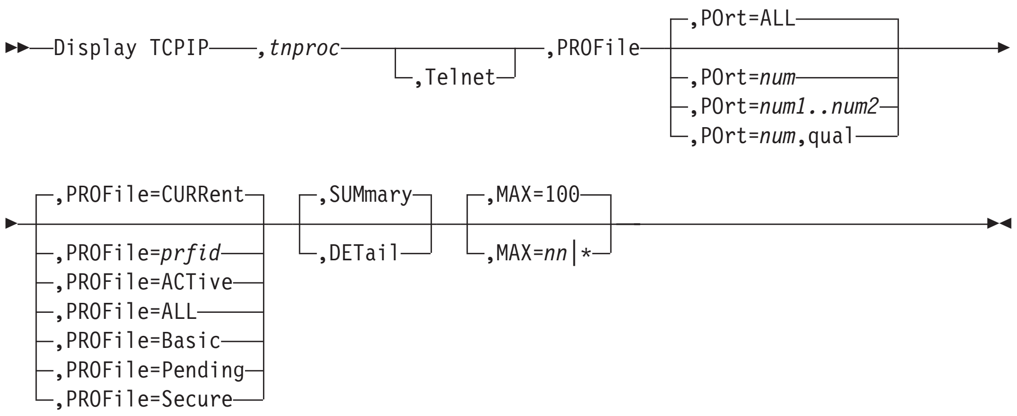
DISPLAY Telnet PROFILE command

Use the PROFILE DISPLAY command to determine:

- Which profile-wide options are in effect for each profile
- Which profiles are still being used
- How many users are on each profile

Format:

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report



Parameters:

tnproc
: The member name of the cataloged procedure that is used to start the Telnet address space.

Telnet
: Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

PROFile
: The profile keyword.

POrt=ALL|*num*|*num1..num2*|*num*,qual
: Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), qualified port (*num*,**qual**) be displayed. **ALL** is the default.

PROFile =CURRent|*prfid*|ACTive|ALL|Basic|Pending|Secure
: The type of profile to display.

- CURRent is the name of the current profile. This is the default.
- *prfid* is the profile ID.
- ACTive is all the active profiles.
- ALL is all profiles, both active and inactive.
- Secure is the secure profiles.
- Pending is the profile waiting for LUNS acknowledgement to become the active profile.
- Basic is the basic profile.

SUMmary|DETail
: SUMmary indicates which parameters are set and the total number of users that are associated with the profile. DETail indicates whether the default value or a configured value is used and the value of each parameter.

MAX=100|*nn*|*
: The number of output lines that are displayed. Valid values are in the range 2-65533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65533 output lines is displayed.

Examples:

To display Telnet summary or detailed profile information, issue the following commands:

```
D TCPIP,TELNET,PROF,PORT=23
EZZ6060I TELNET PROFILE DISPLAY
  PERSIS    FUNCTION        DIA   SECURITY     TIMERS    MISC
 (LMTGCAK) (OPATSKTQSSHRT) (DRF) (PCKLECXN23) (IPKPSTS) (SMLT)
  -------   -------------   ---   ----------   --------   ----
  LMRRCPK   OPATSBTQ*SH*T   ++F   BB*****N**  IPKPST*   SML*
----- PORT: 23 ACTIVE PROF: CURR CONNS: 0
------------------------------------------------------------
   TRANSFORM ACTIVE ON PORT 326
   FORMAT SHORT
   SMFPROFILE    GROUPDETAIL
   TCPIPJOBNAME TCP
   TNSACONFIG ENABLED
   AGENT 161
   CACHETIME 30
   COMMUNITY public
   NOTNSATRACE
   DEBUG TASK EXCEPTION CONSOLE
   DEBUG CONFIG EXCEPTION CONSOLE
   DEBUG CONFIG TRACEOFF
19 OF 19 RECORDS DISPLAYED
```

```
    D TCPIP,TELNET,PROF,PORT=23,DETAIL
(C) EZZ6080I TELNET PROFILE DISPLAY
(L)  PERSIS   FUNCTION       DIA  SECURITY    TIMERS   MISC
(L) (LMTGCAK)(OPATSKTQSSHRT)(DRF)(PCKLECXN23)(IPKPSTS)(SMLT)
     -------  -------------  ---  ----------  -------  ----
     *******  ***TSBTQ***RT  EC*  BB********  *P**STS  *DD* *DEFAULT
     -------  -----BT---HRT  DJ-  ---L---*--  -------  S--- *TGLOBAL
     LM-R-P-  -P---BT---HRT  ---  -B-*------  ----ST-  ---- *TPARMS
     LM*R*P*  *P*TSBTQ**HRT  DJ*  BB********  *P**STS  SDD* CURR
    PERSISTENCE
      LUSESSIONPEND
      MSG07
      NO TKOSPECLU
      TKOGENLURECON           2 NOKEEPONTMRESET
      NOCHECKCLIENTCONN
      DROPASSOCPRINTER
      KEEPLU                  0 (OFF)
    FUNCTIONS
      NOOLDSOLICITOR
      PASSWORDPHRASE
      NOSINGLEATTN
      TN3270E
      SNAEXTENT
      UNLOCKKEYBOARD BEFOREREAD
      UNLOCKKEYBOARD TN3270BIND
      SEQUENTIALLU
      NOSIMCLIENTLU
      HNLOOKUP
      REFRESHMSG10
      NOSHAREACB
      TELNETDEVICE    IBM-3277          D4B32782,**N/A**
      TELNETDEVICE    IBM-3278-2-E      NSX32712,SNX32722  0,0
      TELNETDEVICE    IBM-3278-2        D4B32782,SNX32702
      TELNETDEVICE    IBM-3278-3-E      NSX32702,SNX32703
      TELNETDEVICE    IBM-3278-3        D4B32783,SNX32703
      TELNETDEVICE    IBM-3278-4-E      NSX32702,SNX32704
      TELNETDEVICE    IBM-3278-4        D4B32784,SNX32704
      TELNETDEVICE    IBM-3278-5-E      NSX32702,SNX32705
      TELNETDEVICE    IBM-3278-5        D4B32785,SNX32705
      TELNETDEVICE    IBM-3279-2-E      NSX32702,SNX32702
      TELNETDEVICE    IBM-3279-2        D4B32782,SNX32702
      TELNETDEVICE    IBM-3279-3-E      NSX32702,SNX32703
      TELNETDEVICE    IBM-3279-3        D4B32783,SNX32703
```

```
   TELNETDEVICE    IBM-3279-4-E      NSX32702,SNX32704
   TELNETDEVICE    IBM-3279-4        D4B32784,SNX32704
   TELNETDEVICE    IBM-3279-5-E      NSX32702,SNX32705
   TELNETDEVICE    IBM-3279-5        D4B32785,SNX32705
   TELNETDEVICE    LINEMODE          INTERACT,**N/A**
   TELNETDEVICE    IBM-DYNAMIC       D4C32XX3,D4C32XX3
   TELNETDEVICE    IBM-3287-1        **N/A** ,D6328904
   TELNETDEVICE    TRANSFORM         D4B32782,**N/A**
 DIAGNOSTICS
   DEBUG CONN        DETAIL          CONSOLE
   DEBUG CONN        TRACEOFF
   DEBUG ROUTING     JOBLOG
   NOFULLDATATRACE
 SECURITY
   BASICPORT
   CONNTYPE          BASIC
   KEYRING           NONE
   CRLLDAPSERVER     NONE
   ENCRYPTION        NONE
   CLIENTAUTH        NONE
   NOEXPRESSLOGON
   NONACUSERID
   NOSSLV2
   NOSSLV3
 TIMERS
   INACTIVE                0 (OFF)
   PROFILEINACTIVE      1800
   KEEPINACTIVE            0 (OFF)
   PRTINACTIVE             0 (OFF)
   SCANINTERVAL         3000
   TIMEMARK            12000
   SSLTIMEOUT              5
 MISCELLANEOUS
   SMF
     SMFINIT                 0 (OFF)
     SMFTERM                21
     SMFINIT           TYPE119
     SMFTERM         NOTYPE119
   MAX LIMITS
     MAXRECEIVE          65536
     MAXVTAMSENDQ           50
     MAXTCPSENDQ        999999
     MAXREQSESS             20
     MAXRUCHAIN              0 (OFF)
   LINEMODE
     NOBINARYLINEMODE
     SGA
     CODEPAGE          ISO8859-1     IBM-1047
   TRANSFORM
     NODBCSTRANSFORM
     NODBCSTRACE
----- PORT:   23   ACTIVE               PROF: CURR CONNS:      0
------------------------------------------------------------
FORMAT            LONG
SMFPROFILE        GROUPDETAIL
TCPIPJOBNAME      NO AFFINITY
TNSACONFIG        DISABLED
DEBUG TASK DETAIL CONSOLE
DEBUG CONFIG EXCEPTION CONSOLE
DEBUG CONFIG TRACEOFF
98 OF 98 RECORDS DISPLAYED
```

DISPLAY TCPIP,*tnproc*,STOR

Use the DISPLAY TCPIP,*tnproc*,STOR command to display TN3270E Telnet server (Telnet) storage usage information. You can use this command to verify the load module service level.

To verify load module service level, ensure that the eyecatcher for the module matches the latest PTF service for the module. When you contact IBM Service, you can use this command to verify that you are running on the correct Telnet service level.

Format:

```
►►──Display ──TCPIP,──tnproc──,STOR──┬──────────────────┬──────────────────────────►◄
                                      └─MODule=mod_name─┘
```

Parameters:

STOR

Requests storage information.

If no other option is specified, the command displays the current and maximum storage usage for Telnet and any Telnet storage limits. The maximum storage usage is the highest amount of storage that Telnet has used since it started. See message EZZ8453I in z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM) for a description of the output displayed.

MODULE

Displays the load module name that contains the module, module address and the first 48 bytes of storage.

This command displays modules within load modules EZBTNINI, EZBTMCTL, EZBTPGUE, EZBTTMST, and EZBTZMST for Telnet.

Examples:
To display Telnet storage usage, issue the following command:

```
d tcpip,tn3270,stor
EZZ8453I TELNET STORAGE
EZZ8454I TN3270   STORAGE            CURRENT   MAXIMUM     LIMIT
EZD2018I 31-BIT
EZZ8455I          ECSA                   85K      137K   NOLIMIT
EZZ8455I          PRIVATE              6810K     7241K   NOLIMIT
EZD2018I 64-BIT
EZZ8455I          TRACE HVPRIVATE      1025M     1025M     1025M
EZZ8459I DISPLAY TELNET STOR COMPLETED SUCCESSFULLY
```

DISPLAY TCPIP,TELNET

Use the DISPLAY TCPIP,TELNET command from the MVS operator console to display the name, version, and status of the TN3270E Telnet servers (Telnet) that are or were running.

Format:

```
►►──Display TCPIP──,TELNET─────────────────────────────────────────────────────────►◄
```

Parameters:
There are no parameters.

Examples:
To view the name, version, and status of Telnet servers, issue the following command:

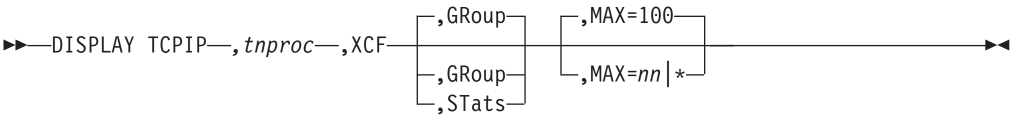
```
d tcpip,telnet

EZAOP60I TELNET STATUS REPORT
```

```
TELNET NAME  VERSION    STATUS
-----------  --------   -------------------------------
TELNET       CS V2R1   ACTIVE
TELNET5      CS V2R1   INACTIVE (STOP CMD)
TELNET4      CS V2R1   INACTIVE (STOP CMD)
*** END TELNET STATUS REPORT ***
```

DISPLAY TCPIP,*tnproc*,XCF

Use the DISPLAY TCPIP,*tnproc*,XCF command from an operator console to request TN3270E Telnet server XCF information. You must specify the Telnet procedure name.

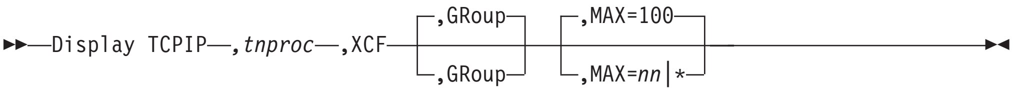
Format:



The following descriptions provide details of the DISPLAY TCPIP,*tnproc*,XCF commands that you can use.

DISPLAY TCPIP,*tnproc*,XCF<,GRoup> command:
Use the Display TCPIP,*tnproc*,XCF<,GRoup> command to see the state and status of all the Telnet members of the XCF group.

Format:



Parameters:

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

XCF
The XCF keyword.

GRoup
The type of XCF information to display. Use the GRoup parameter to display status information for all XCF Telnets in the XCF group.

MAX=100|*nn*|*
The number of output lines that are displayed. Valid values are in the range 2 - 65 533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65 533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65 533 output lines is displayed.

Example:

```
D TCPIP,TLUNS1,XCF
EZZ6089I TLUNS1 XCF GROUP DISPLAY
GROUP NAME: EZZTLUNS CONNECTTIMEOUT:        90
XCFMONITOR:       10 RECOVERYTIMEOUT:       80
LUNS LISTENER: 192.168.17.2..8000
```

```
                            LUNS--------------- LUNR----------
MVSNAME  TNNAME   PDMON CTR RANK STATE   STATUS STATE   STATUS
-------- -------- ----- --- ------------------- --------------
RANS17   TLUNR1          12                     ACTIVE     L
RANS17   TLUNR2          12                     ACTIVE    CP
RANS17   TLUNR3          12                     ACTIVE     R
RANS17   TLUNS1          12 P101 STANDBY        STANDBY
RANS17   TLUNS2          12 P100 STANDBY        STANDBY
RANS18   TLUNR1          12                     ACTIVE     L
RANS18   TLUNR2          12                     ACTIVE     L
RANS18   TLUNR3          12                     ACTIVE
RANS18   TLUNS1          12 P101 STANDBY        STANDBY
RANS19   TLUNRA          12                     ACTIVE     L
RANS19   TLUNR1          12                     STANDBY
RANS19   TLUNR2          12                     STANDBY
RANS19   TLUNR3          12                     STANDBY
RANS19   TLUNS1          12 P101 ACTIVE   R     STANDBY
22 OF 22 RECORDS DISPLAYED
```

Display TCPIP,*tnproc*,XCF,STats command:

Use the Display TCPIP,*tnproc*,XCF,STats command to see the performance statistics of the LUNR and LUNS.

Format:

```
                                            ┌─,MAX=100──┐
►►──Display TCPIP──,tnproc──,XCF──,STats────┼───────────┼──────────────────────►◄
                                            └─,MAX=nn|*─┘
```

Parameters:

tnproc
: The member name of the cataloged procedure that is used to start the Telnet address space.

XCF
: The XCF keyword.

STats
: The type of XCF information to display. The STats parameter displays performance statistics for all XCF Telnets in the XCF group.

MAX=100|*nn*|*****
: The number of output lines that are displayed. Valid values are in the range 2 - 65 533. The default value is 100. An asterisk (*) indicates that all output lines are displayed. The command can display a maximum of 65 533 output lines (control, label, and data lines). Therefore, if you specify an asterisk (*), a maximum of 65 533 output lines is displayed.

Example:

The following example displays possible output from this command.

```
D TCPIP,TLUNS1,XCF,ST
EZZ6088I TLUNS1 XCF STATS DISPLAY
    INTERVAL: 60S          PEND       RECV          SEND
 NEXT UPDATE:  9S    RTT    RCRD    TIME  RCRD    TIME  RCRD
====PARTNERS=====
RANS17   TLUNR1    -----   -----   ----- -----   ----- -----
            LAST:   250M      0     616U    6     413U    6
             AVG:   154M      0       2M   34     575U   13
```

```
RANS17   TLUNR2    -----   -----   ----- -----   ----- -----
            LAST:     1M      0     273U    4     393U    4
             AVG:     1M      0       1M   43     500U   21
RANS17   TLUNR3    -----   -----   ----- -----   ----- -----
            LAST:     1M      0      37M   2K       8M  831
             AVG:   965U      0       4M  179       1M   89
RANS18   TLUNR1    -----   -----   ----- -----   ----- -----
            LAST:   236M      0     629U    6     465U    6
             AVG:   127M      0       3M   83     833U   38
RANS18   TLUNR2    -----   -----   ----- -----   ----- -----
            LAST:     1M      0     289U    4     311U    4
             AVG:     2M      0       5M  244       1M  127
RANS18   TLUNR3    -----   -----   ----- -----   ----- -----
            LAST:     1M      0     356U    4     454U    4
             AVG:     1M      0     572U   13     380U    7
RANS19   TLUNRA    -----   -----   ----- -----   ----- -----
            LAST:   318U      0     519U    6     433U    6
             AVG:   493U      0      12M  168       4M   77
26 OF 26 RECORDS DISPLAYED
```

MODIFY command

The MODIFY command allows you to dynamically change the characteristics of an active task. The abbreviated version of the command is the letter F.

This is the general format of MODIFY:

```
►►──┬─MODIFY─┬──procname──,──parameter──────────────────────────────────►◄
    └─F──────┘
```

procname
: The name of the member in a procedure library that was used to start the server or address space.

parameter
: Any of the parameters that are valid for the server.

The following servers or address spaces support the MVS MODIFY command. Not all servers support the same parameters. For further descriptions of the supported parameters, see Table 8.

Table 8. Servers or address spaces that support the MVS Modify command

Server or Address space	Main parameters	Additional information
Automated domain name registration application (EZBADNR)	DEBUG, DISPLAY, REFRESH	“MODIFY command: Automated domain name registration application (EZBADNR)” on page 150
Communications Server SMTP (CSSMTP) application	DISPLAY, FLUSHRETRY, LOGLEVEL, REFRESH, REFRESHIPLIST, REFRESHTARGETS, RESUME, SUSPEND, USEREXIT	“MODIFY command: Communications Server SMTP application (CSSMTP)” on page 167
DCAS	DEBUG	“MODIFY command: DCAS” on page 180
Defense manager daemon (DMD)	DISPLAY, REFRESH, FORCE_INACTIVE	“MODIFY command: Defense Manager daemon” on page 181
FTP server	DUMP, DEBUG	“MODIFY command: FTP” on page 182

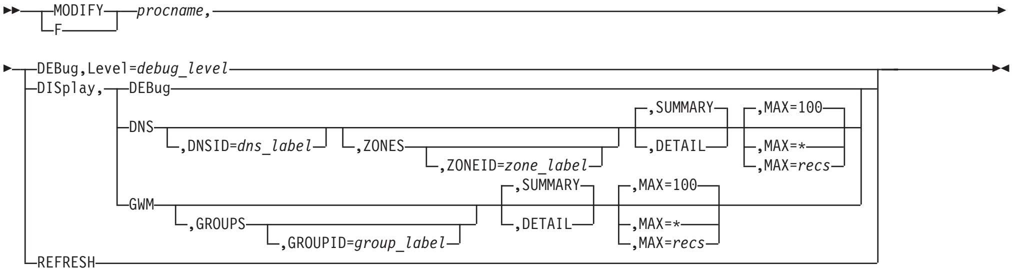
Table 8. Servers or address spaces that support the MVS Modify command (continued)

Server or Address space	Main parameters	Additional information
IKE server	DISPLAY, REFRESH	“MODIFY command: IKE server” on page 187
Load Balancing Advisor	DEBUG, DISPLAY	“MODIFY command: z/OS Load Balancing Advisor” on page 228
Load Balancing Agent	DEBUG, DISPLAY, QUIESCE, ENABLE	“MODIFY command: z/OS Load Balancing Agent” on page 236
NCPROUTE server	C, PARMS, PROFILE, QUERY, GATEWAYS, TABLES	“MODIFY command: NCPROUTE” on page 188
Network security services server	DISPLAY, REFRESH	“MODIFY command: Network security services server” on page 190
OMPROUTE	KILL, RECONFIG, ROUTESA, OSPF, RIP, GENERIC, RTTABLE, IPV6OSPF, IPV6RIP, GENERIC6, RT6TABLE, TRACE, DEBUG, TRACE6, DEBUG6, SADEBUG	“MODIFY command: OMPROUTE” on page 192
Policy Agent	LOGLEVEL, TRACE, DEBUG, QUERY, REFRESH, MEMTRC, SRVLSTN, UPDATE, MON	“MODIFY command: Policy Agent” on page 205
Resolver address space	DISPLAY, FLUSH, REFRESH	“MODIFY command: Resolver address space” on page 209
REXEC	EXIT, TSOPROC, MSGCLASS, TSCLASS, TRACE, PURGE	“MODIFY command: REXEC” on page 213
Rpcbind server	TRACE	“MODIFY command: RPCBIND” on page 213
SMTP	SMSG	“MODIFY command: SMTP” on page 214
SNALINK LU0	HALT	“MODIFY command: SNALINK LU0” on page 219
SNALINK LU6.2	CANCEL, DROP, HALT, LIST, RESTART, TRACE	“MODIFY command: SNALINK LU 6.2” on page 220
SNMP agent	INTERVAL, TRACE	“MODIFY command: SNMP agent” on page 223
SNMP network SLAPM2 subagent	DEBUG, CACHE, QUERY	“MODIFY command: SNMP Network SLAPM2 subagent” on page 224
Syslog daemon	ARCHIVE, DISPLAY, RESTART	“MODIFY command: Syslog Daemon” on page 224
TNF	DISPLAY, REMOVE	“MODIFY command: VMCF and TNF” on page 227
Trap forwarder daemon	QUERY, REFRESH, TRACE	“MODIFY command: Trap forwarder daemon (TRAPFWD)” on page 226
VMCF	DISPLAY, REMOVE	“MODIFY command: VMCF and TNF” on page 227
X.25 NPSI server	CANCEL, DEBUG, EVENTS, HALT, LIST, RESTART, SNAP, TRACE, TRAFFIC	“MODIFY command: X.25 NPSI server” on page 227

MODIFY command: Automated domain name registration application (EZBADNR)

Use the MODIFY command to control the automated domain name registration (ADNR) application from the operator's console.

Format



Parameters

procname
: The member name of the cataloged procedure that is used to start the automated domain name registration application.

DEBug,Level=*debug_level*
: Changes the automated domain name registration application debug level. See Automated domain name registration application (EZBADNR) configuration file in the z/OS Communications Server: IP Configuration Reference for details on valid automated domain name registration application debug levels.

DISplay,DEBug
: Displays the automated domain name registration application debug level including the active individual logging levels.

DISplay,DNS[,DNSID=*dns_label***][,SUMMARY][,MAX=***recs***]**
: Displays a summary of Domain Name System (DNS) information for the name server specified by the *dns_label* value or for all configured name servers. All configured name servers are displayed if the DNSID parameter is not specified. If the DNSID parameter is specified, the *dns_label* value must match the *dns_label* value used on one of the dns statements in the automated domain name registration application configuration file. See the following summary DNS information:

- DNS label
- DNS status

The number of name servers displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, then all name servers are displayed.

DISplay,DNS[,DNSID=*dns_label***],DETAIL[,MAX=***recs***]**
: Displays detailed DNS information for the name server specified by the *dns_label* value or for all configured name servers. All configured name servers are displayed if the DNSID parameter is not specified. If the DNSID parameter is specified, the *dns_label* value must match the *dns_label* value used on one of the dns statements in the automated domain name registration application configuration file. See the following detailed DNS information:

- DNS label

- DNS status
- DNS IP address and port
- Number of zones defined
- Number of zones active

The number of name servers displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, then all name servers are displayed.

DISplay,DNS[,DNSID=*dns_label***],ZONES[,ZONEID=***zone_label***][,SUMMARY][,MAX=***recs***]**

Displays a summary of zone information for the zone specified by the *zone_label* value or for all zones.

- All zones are displayed if DNSID and ZONEID parameters are not specified.
- All zones under a specific configured name server are displayed if the DNSID parameter is specified and the ZONEID parameter is not specified.
- If the DNSID parameter is specified, the *dns_label* value must match the *dns_label* value used on one of the dns statements in the automated domain name registration application configuration file.
- If ZONEID and DNSID parameters are specified, the *zone_label* value must match the *zone_label* value on one of the zone parameters on the dns statement with label *dns_label* in the automated domain name registration application configuration file.
- If the ZONEID parameter is specified and the DNSID parameter is not specified, the *zone_label* value must match the *zone_label* value on one of the zone parameters that is on one of the dns statements in the automated domain name registration configuration file. Only information about the zone specified by the ZONEID parameter and the name server that contains the zone is displayed.

See the following summary zone information:

- DNS label
- DNS status
- Zone information

For each zone the following information is displayed:

- Zone label
- Zone status

The number of zones displayed is limited by the MAX=*recs* parameter. When this maximum is reached, no more zones or name servers are displayed. The default value is 100. If MAX=* is specified, then all zones are displayed.

DISplay,DNS[,DNSID=*dns_label***],ZONES[,ZONEID=***zone_label***],DETAIL[,MAX=***recs***]**

Displays detailed zone information for the zone specified by the *zone_label* value or for all zones.

- All zones are displayed if DNSID and ZONEID parameters are not specified.
- All zones under a specific configured name server are displayed if the DNSID parameter is specified and the ZONEID parameter is not specified.
- If the DNSID parameter is specified, the *dns_label* value must match the *dns_label* value on one of the dns statements in the automated domain name registration application configuration file.

- If the ZONEID and DNSID parameters are specified, the *zone_label* value must match the *zone_label* value on one of the zone parameters on the dns statement with label *dns_label* in the automated domain name registration application configuration file.
- If the ZONEID parameter is specified and the DNSID parameter is not specified, the *zone_label* value must match the *zone_label* value on one of the zone parameters that is on one of the dns statements in the automated domain name registration application configuration file. Only information about the zone specified by the ZONEID parameter and the name server that contains the zone is displayed.

See the following detailed zone information:

- DNS label
- DNS status
- DNS IP address and port
- Number of zones defined
- Number of zones active
- Zone information

For each zone, the following information is displayed:

- Zone label
- Zone status
- Status timestamp
- Domain suffix
- TSIG flags
- DNS resource record information

For each DNS resource record, the following information is displayed:

- Label
- Status
- TTL
- Class
- Type
- IP address
- GWM label
- Group label
- Last update timestamp

The number of zones displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, then all zones are displayed.

`DISplay,GWM[,SUMMARY]`

Displays a summary of the Global Workload Manager (GWM) information. See the following summary GWM information:

- GWM label
- GWM status

`DISplay,GWM,DETAIL`

Displays detailed GWM information. See the following detailed GWM information:

- GWM label
- GWM status

- Status timestamp
- GWM IP address and port
- Host (local) IP address
- Universally unique identifier (UUID)
- Update interval
- Last update timestamp

DISplay,GWM,GROUPS[,GROUPID=*group_label*][,SUMMARY][,MAX=*recs*]

Displays a summary of group information for the group specified by the *group_label* value or for all groups. All groups are displayed if the GROUPID parameter is not specified. If the GROUPID parameter is specified, the *group_label* value must match the *host_group_label* value on one of the host_group statements or the *server_group_label* value on one of the server_group statements in the automated domain name registration application configuration file. See the following summary group information:

- GWM label
- GWM status
- Group information

For each group, the following information is displayed:

- Group label
- Group name

The number of groups displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, then all groups are displayed.

DISplay,GWM,GROUPS[,GROUPID=*group_label*],DETAIL[,MAX=*recs*]

Displays detailed group information for the group specified by the *group_label* value or for all groups. All groups are displayed if the GROUPID parameter is not specified. If the GROUPID parameter is specified, the *group_label* value must match the *host_group_label* value on one of the host_group statements or the *server_group_label* value on one of the server_group statements in the automated domain name registration application configuration file.

See the following detailed group information:

- GWM label
- GWM status
- Status timestamp
- GWM IP address and port
- Host (local) IP address
- Universally unique identifier (UUID)
- Update interval
- Last update timestamp
- Group information

For each group, the following information is displayed:

- Group label
- Group name
- Group type
- DNS label
- Zone label
- Member information

For each member, the following information is displayed:

- Member *hostname* (if available)
- IP address information

For each member IP address, the following information is displayed:

- IP address and port
- Protocol, if the member is part of a server group. Protocol is not displayed if the member is part of a host group.
- Status
- Flags
- Update count

The number of groups displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, all groups are displayed.

REFRESH

Initiates a dynamic reconfiguration using the configuration file defined in the cataloged procedure that is used to start the automated domain name registration application. This causes the automated domain name registration application to resynchronize all dynamic DNS zones with the modified configuration. DNS records representing prior configuration elements existing in the previous configuration are removed.

While the new configuration file is being processed, the existing debug level is used, regardless of how it was set (using the last configuration file or with the MODIFY DEBUG command). After the new configuration file has been successfully processed, the value specified on the debug_level statement of the new configuration file takes effect. If the debug_level statement is not specified in the new configuration file, the debug level defaults to a level of 7 (ERROR, WARNING, EVENT). If the new configuration file contains errors that cause it to be rejected, the debug level that was in effect prior to the dynamic reconfiguration is used.

Rule: When you update the arm_element_suffix statement, perform the following steps to ensure that the ADNR application is able to automatically restart:

1. Add the new element name to the ARM policy and add or change the arm_element_suffix value in the ADNR configuration file.
2. Refresh the ADNR application.
3. Optionally, remove the old element name from the ARM policy.

Example 1: The MODIFY DISPLAY DNS command summarizes all name servers that are managed by the automated domain name registration application.

```
F ADNR,DIS,DNS
EZD1254I DNS SUMMARY
DNS LABEL        : DNS2
 DNS STATUS      : ACTIVE
DNS LABEL        : DNS7
 DNS STATUS      : DELETING
2 of 2 RECORDS DISPLAYED
```

DNS LABEL

The DNS label configured in the automated domain name registration application configuration file on the dns statement.

DNS STATUS

The status of the DNS server. The following list shows the possible values:

ACTIVE
The automated domain name registration application is operating under normal conditions.

DELETED
The automated domain name registration application has successfully deleted the name server and its subordinate zones from its configuration following a MODIFY REFRESH command. The automated domain name registration application is waiting for zones under other name servers to be deleted.

DELETING
The automated domain name registration application is in the process of deleting the subordinate zones and resource records.

INITIAL
The automated domain name registration application has not yet started managing data for the name server. This occurs while the automated domain name registration application is initializing or shortly after dynamic reconfiguration has begun.

SHUTTING_DOWN
The automated domain name registration application is terminating.

Example 2: The MODIFY DISPLAY DNS DETAIL command provides details for all name servers that are managed by the automated domain name registration application.

```
F ADNR,DIS,DNS,DETAIL
EZD1254I DNS DETAIL
DNS LABEL        : DNS2
 DNS STATUS      : ACTIVE
 DNS IPADDR..PORT: 2001:DB8:10::81:2:2..53
 ZONES DEFINED   : 2
 ZONES ACTIVE    : 2
DNS LABEL        : DNS7
 DNS STATUS      : DELETING
 DNS IPADDR..PORT: 10.81.7.7..53
 ZONES DEFINED   : 1
 ZONES ACTIVE    : 0
2 of 2 RECORDS DISPLAYED
```

DNS LABEL
The DNS label configured in the automated domain name registration application configuration file on the dns statement.

DNS STATUS
The DNS server status.

DNS IPADDR..PORT
The remote IP address and port of the name server.

ZONES DEFINED
The number of zones defined using the zone parameter on the dns statement.

ZONES ACTIVE
The number of active zones.

Example 3: The MODIFY DISPLAY DNS ZONES command supplies zone summary information for all name servers that are managed by the automated domain name registration application.

```
F ADNR,DIS,DNS,ZONES
EZD1254I DNS ZONE SUMMARY
DNS LABEL        : DNS2
 DNS STATUS      : ACTIVE
 ZONE LABEL      : ZONE2
  ZONE STATUS    : SYNCHRONIZED
 ZONE LABEL      : ZONE3
  ZONE STATUS    : SYNCHRONIZED
DNS LABEL        : DNS7
 DNS STATUS      : SHUTTING_DOWN
 ZONE LABEL      : ZONE7
  ZONE STATUS    : DELETING
3 of 3 RECORDS DISPLAYED
```

DNS LABEL

The DNS label configured in the automated domain name registration application configuration file on the dns statement.

DNS STATUS

The DNS server status.

ZONE LABEL

The zone label configured in the automated domain name registration application configuration file using the zone parameter on the dns statement.

ZONE STATUS

The status of the zone. The following list shows the possible values:

DELETED

The zone managed by the automated domain name registration application has been terminated.

DELETING

The zone managed by the automated domain name registration application is being terminated. A zone delete is in progress.

INITIAL

The automated domain name registration application has not yet started managing data for the zone. This occurs while the automated domain name registration application is initializing, or shortly after dynamic reconfiguration has begun.

NOT_RESPONSIVE_ZONE_UPDATE_PENDING

The zone managed by the automated domain name registration application is not responsive. Dynamic update probes are periodically sent to the zone in this state until one is successful.

NOT_RESPONSIVE_ZONE_XFER_PENDING

The zone managed by the automated domain name registration application is not responsive. A zone transfer is in progress.

RESYNCH_ZONE_UPDATE_PENDING

The zone managed by the automated domain name registration application is being resynchronized. A zone update is in progress. Resynchronization occurs during initialization or dynamic reconfiguration of the automated domain name registration application.

RESYNCH_RECONCILE_PENDING

The zone managed by the automated domain name registration application is being resynchronized. A reconcile of the zone is in progress. Resynchronization occurs during initialization or dynamic reconfiguration of the automated domain name registration application. A zone can remain in this state indefinitely if one of the following condition is true:

- The GWM is not active
- No groups are defined to the automated domain name registration application
- No groups reference the zone

RESYNCH_ZONE_XFER_PENDING
The zone managed by the automated domain name registration application is being resynchronized. A zone transfer is in progress. Resynchronization occurs during initialization or dynamic reconfiguration of the automated domain name registration application.

SHUTTING_DOWN
The automated domain name registration application is terminating.

SYNCHRONIZED
The automated domain name registration application is in synch with the name server and is able to update the zone.

Example 4: The MODIFY DISPLAY DNS ZONES DETAIL command supplies zone detail information about all name servers that are managed by the automated domain name registration application.

```
F ADNR,DIS,DNS,ZONES,DETAIL
EZD1254I DNS ZONE DETAIL
DNS LABEL        : DNS2
 DNS STATUS      : ACTIVE
 DNS IPADDR..PORT: 2001:DB8:10::81:2:2..53
 ZONES DEFINED   : 2
 ZONES ACTIVE    : 2
 ZONE LABEL      : ZONE2
  ZONE STATUS    : SYNCHRONIZED
  DOMAIN SUFFIX  : ZONE2.MYCORP.COM
  ZONE TIMESTAMP : 04/27/2005 12:31:16
  TSIG FLAGS     : TRANSFER UPDATE
  DNS RR LABEL   : FTP
   DNS RR STATUS : PRESENT
   TTL           : 2147483647
   CLASS         : IN
   TYPE          : AAAA
   RDATA         : 2001:0DB8:10::81:2:2
   GWM LABEL     : GWM1
   GROUP LABEL   : FTP_GROUP
   LAST UPDATE   : 04/27/2005 05:25:21
 ZONE LABEL      : ZONE3
  DOMAIN SUFFIX  : ZONE3.MYCORP.COM
  ZONE STATUS    : SYNCHRONIZED
  ZONE TIMESTAMP : 04/27/2005 05:25:22
  TSIG FLAGS     :
  DNS RR LABEL   : FTP3
   DNS RR STATUS : UPDATE-ADD_IN_PROGRESS
   TTL           : 0
   CLASS         : IN
   TYPE          : A
   RDATA         : 10.81.3.3
   GWM LABEL     : GWM1
   GROUP LABEL   : FTP_GROUP
   LAST UPDATE   : 04/27/2005 04:17:31
  DNS RR LABEL   : FTP3
   DNS RR STATUS : NOT_PRESENT
   TTL           : 86400
   CLASS         : IN
   TYPE          : AAAA
   RDATA         : 2001:DB8:10::81:3:3
   GWM LABEL     : GWM1
   GROUP LABEL   : FTP_GROUP
   LAST UPDATE   : 04/27/2005 04:17:33
DNS LABEL       : DNS7
 DNS STATUS      : SHUTTING_DOWN
 DNS IPADDR..PORT: 10.81.7.7..53
 ZONES DEFINED   : 1
 ZONES ACTIVE    : 0
 ZONE LABEL      : ZONE7
  DOMAIN SUFFIX  : ZONE7.MYCORP.COM
  ZONE STATUS    : DELETING
  ZONE TIMESTAMP : 04/27/2005 02:54:00
  TSIG FLAGS     :
  DNS RR LABEL   : FTP7
   DNS RR STATUS : UPDATE-DEL_IN_PROGRESS
   TTL           : 86400
   CLASS         : IN
   TYPE          : AAAA
   RDATA         : 2001:DB8:10::81:7:7
   GWM LABEL     : GWM1
   GROUP LABEL   : HOST_GROUP
   LAST UPDATE   : 04/27/2005 03:10:15
3 of 3 RECORDS DISPLAYED
```

DNS LABEL
The DNS label configured in the automated domain name registration application configuration file on the dns statement.

DNS STATUS
The DNS server status.

DNS IPADDR..PORT
The remote IP address and port of the name server.

ZONES DEFINED
The number of zones defined using the zone parameter on the dns statement.

ZONES ACTIVE
The number of active zones.

ZONE LABEL
The zone label configured in the automated domain name registration application configuration file using the zone parameter on the dns statement.

ZONE STATUS
The status of the zone. The following list shows the possible values:

DELETED
The zone managed by the automated domain name registration application has been terminated.

DELETING
The zone managed by the automated domain name registration application is being terminated. A zone delete is in progress.

INITIAL
The automated domain name registration application has not yet started managing data for the zone. This occurs while the automated domain name registration application is initializing, or shortly after dynamic reconfiguration has begun.

NOT_RESPONSIVE_ZONE_UPDATE_PENDING
The zone managed by the automated domain name registration application is not responsive. A zone update is in progress.

NOT_RESPONSIVE_ZONE_XFER_PENDING
The zone managed by the automated domain name registration application is not responsive. A zone transfer is in progress.

RESYNCH_ZONE_UPDATE_PENDING
The zone managed by the automated domain name registration application is being resynchronized. A zone update is in progress. Resynchronization occurs during initialization or dynamic reconfiguration of the automated domain name registration application.

RESYNCH_RECONCILE_PENDING
The zone managed by the automated domain name registration application is being resynchronized. A reconcile of the zone is in progress. Resynchronization occurs during initialization or dynamic reconfiguration of the automated domain name registration application. A zone can remain in this state indefinitely if one of the following condition is true:

- The GWM is not active
- No groups are defined to ADNR
- No groups reference the zone

RESYNCH_ZONE_XFER_PENDING
The zone managed by the automated domain name registration application is being resynchronized. A zone transfer is in progress. Resynchronization occurs during initialization or dynamic reconfiguration of the automated domain name registration application.

SHUTTING_DOWN
The automated domain name registration application is terminating.

SYNCHRONIZED
The automated domain name registration application is synchronized with the name server and is able to update the zone.

DOMAIN SUFFIX
The domain suffix of the zone for which the name server is authoritative.

ZONE TIMESTAMP
The timestamp in UTC format specifying when the DNS server reached the status indicated by the ZONE STATUS value.

TSIG FLAGS
An indication of whether DNS transactions are signed. The following list shows the possible flag values:

TRANSFER
DNS transfers are signed.

UPDATE
DNS updates are signed.

If no flags are displayed, then DNS transactions are not signed.

DNS RR LABEL
The DNS resource record label.

DNS RR STATUS
The DNS resource record status. The following list shows the possible status values:

NOT_PRESENT
The DNS resource record is not currently present in the name server. This indicates that the host or application is not available for one of the following reasons:

- The IP address was not found by the GWM.
- The IP address was found by the GWM but an application was not found to be listening on the specific port.
- The host or application has been quiesced.

PRESENT
The DNS resource record is currently present in the name server. This indicates that the host or application is available.

UPDATE-ADD_IN_PROGRESS
The DNS resource record is being added to the name server.

UPDATE-DEL_IN_PROGRESS
The DNS resource record is being deleted from the name server.

REPLACE_IN_PROGRESS
The DNS resource record is being replaced as a result of a TTL change.

TTL
The time to live value in seconds associated with this DNS record in the name server.

CLASS
The DNS record class always has the value INTERNET, which is abbreviated as IN.

TYPE
The DNS record type. Possible values are:

A Designates IPv4.

AAAA
Designates IPv6.

RDATA
The DNS record data.

- RDATA is an IPv4 address when TYPE is A.
- RDATA is an IPv6 address when TYPE is AAAA.

GWM LABEL
The GWM label configured in the automated domain name registration application configuration file on the gwm statement.

GROUP LABEL
The group label configured in the automated domain name registration application configuration file on the host_group statement or the server_group statement.

LAST UPDATE
The timestamp, in UTC format, specifying the most recent update by ADNR for this DNS record; N/A is displayed if ADNR has never sent an update for this record to the name server.

Example 5: The MODIFY DISPLAY GWM command summarizes the state of the GWM.

```
F ADNR,DIS,GWM
EZD1254I GWM SUMMARY
GWM LABEL        : GWM1
 GWM STATUS      : GWM_ACTIVE
1 of 1 RECORDS DISPLAYED
```

GWM LABEL
The GWM label configured in the automated domain name registration application configuration file on the gwm statement.

GWM STATUS
The status of the GWM advising the automated domain name registration application. Possible values are:

CONNECTED
The automated domain name registration application is connected to the GWM.

CONVERGENCE_PENDING
The automated domain name registration application is waiting a fixed period of time for information about all configured groups to be returned from the GWM.

DISCONNECTED
The automated domain name registration application is not connected to the GWM.

GETWEIGHTS_RSP_PENDING
The automated domain name registration application is waiting for a SASP GetWeights response message from the GWM.

GWM_ACTIVE
The state of the GWM after it has exited the CONVERGENCE_PENDING state. This is the normal steady state of the GWM. When the GWM is in this state, all changes in the status of any configured group are received by the automated domain name registration application and forwarded to the appropriate name servers. The GWM remains in this state until there is a configuration change or until either the GWM or the ADNR application is stopped.

PRE_REG_DEREGISTRATION_RSP_PENDING
The automated domain name registration application is waiting for a SASP DeRegistration response message from the GWM as a result of GWM communication initialization.

REGISTRATION RSP_PENDING
The automated domain name registration application is waiting for a SASP Registration response message from the GWM.

SETLBSTATE RSP_PENDING
The automated domain name registration application is waiting for a SASP SetLoadBalancerState response message from the GWM.

SHUTTING_DOWN
The automated domain name registration application is terminating.

Example 6: The MODIFY DISPLAY GWM DETAIL command provides details about the GWM.

```
F ADNR,DIS,GWM,DETAIL
EZD1254I GWM DETAIL
GWM LABEL        : GWM1
 GWM STATUS      : GWM_ACTIVE
 GWM TIMESTAMP   : 04/27/2005 12:32:01
 GWM IPADDR..PORT: 10.81.1.1..3860
 LOCAL IPADDR    : 10.81.4.4
 UUID            : UUID1
 UPDATE INTERVAL : 60
 LAST UPDATE     : 04/27/2005 01:05:03
1 of 1 RECORDS DISPLAYED
```

GWM LABEL
The GWM label configured in the automated domain name registration application configuration file on the gwm statement.

GWM STATUS
The status of the GWM advising the automated domain name registration application. Possible values are:

CONNECTED
The automated domain name registration application is connected to the GWM.

CONVERGENCE_PENDING
The automated domain name registration application is waiting a fixed period of time for information about all configured groups to be returned from the GWM.

DISCONNECTED
The automated domain name registration application is not connected to the GWM.

GETWEIGHTS_RSP_PENDING
The automated domain name registration application is waiting for a SASP GetWeights response message from the GWM.

GWM_ACTIVE
The state of the GWM after it has exited the CONVERGENCE_PENDING state. This is the normal steady state of the GWM. When the GWM is in this state, all changes in the status of any configured group are received by the automated domain name registration application and forwarded to the appropriate name servers. The GWM remains in this state until there is a configuration change or until either the GWM or the automated domain name registration application is stopped.

PRE_REG_DEREGISTRATION_RSP_PENDING
The automated domain name registration application is waiting for a SASP DeRegistration response message from the GWM as a result of GWM communication initialization.

REGISTRATION RSP_PENDING
The automated domain name registration application is waiting for a SASP Registration response message from the GWM.

SETLBSTATE RSP_PENDING
The automated domain name registration application is waiting for a SASP SetLoadBalancerState response message from the GWM.

SHUTTING_DOWN
The automated domain name registration application is terminating.

GWM TIMESTAMP
The timestamp in UTC format specifying when the GWM reached the status indicated by the GWM STATUS value.

GWM IPADDR..PORT
The remote IP address and port of the GWM.

LOCAL IPADDR
The local IP address that the automated domain name registration application used to connect to the GWM.

UUID
The universally unique identifier that the automated domain name registration application used to connect to the GWM.

UPDATE INTERVAL
The GWM's update interval in seconds. See the appropriate GWM documentation for more information.

LAST UPDATE
The timestamp, in UTC format, specifying the most recent update (SASP SendWeights message) received from the GWM; N/A is displayed if the GWM has not sent the automated domain name registration application an update since the connection to the GWM became active.

Example 7: The MODIFY DISPLAY GWM GROUPS command supplies group summary information about the GWM.

```
F ADNR,DIS,GWM,GROUPS
EZD1254I GWM GROUP SUMMARY
GWM LABEL        : GWM1
 GWM STATUS      : GWM_ACTIVE
 GROUP LABEL     : FTP_GROUP
  GROUP NAME     : FTP.ZONE2.MYCORP.COM
 GROUP LABEL     : HOST_GROUP
  GROUP NAME     : HOST7.ZONE7.MYCORP.COM
2 of 2 RECORDS DISPLAYED
```

GWM LABEL
The GWM label configured in the automated domain name registration application configuration file on the gwm statement.

GWM STATUS
The status of the GWM advising the automated domain name registration application. Possible values are:

CONNECTED
The automated domain name registration application is connected to the GWM.

CONVERGENCE_PENDING
The automated domain name registration application is waiting a fixed period of time for information about all configured groups to be returned from the GWM.

DISCONNECTED
The automated domain name registration application is not connected to the GWM.

GETWEIGHTS_RSP_PENDING
The automated domain name registration application is waiting for a SASP GetWeights response message from the GWM.

GWM_ACTIVE
The state of the GWM after it has exited the CONVERGENCE_PENDING state. This is the normal steady state of the GWM. When the GWM is in this state, all changes in the status of any configured group are received by the automated domain name registration application and forwarded to the appropriate name servers. The GWM remains in this state until there is a

configuration change or until either the GWM or the automated domain name registration application is stopped.

PRE_REG_DEREGISTRATION_RSP_PENDING
The automated domain name registration application is waiting for a SASP DeRegistration response message from the GWM as a result of GWM communication initialization.

REGISTRATION RSP_PENDING
The automated domain name registration application is waiting for a SASP Registration response message from the GWM.

SETLBSTATE RSP_PENDING
The automated domain name registration application is waiting for a SASP SetLoadBalancerState response message from the GWM.

SHUTTING_DOWN
The automated domain name registration application is terminating.

GROUP LABEL
The group label configured in the automated domain name registration application configuration file on the host_group statement or on the server_group statement.

GROUP NAME
The group name registered with the GWM. The group name is defined in the automated domain name registration application configuration file using the host_group_name parameter on the host_group statement or the server_group_name parameter on the server_group statement concatenated to the domain_suffix of the zone identified by the dns and zone parameters on the host_group statement or the server_group statement.

Example 8: The MODIFY DISPLAY GWM GROUPS DETAIL command supplies group detail information about the GWM.

```
F ADNR,DIS,GWM,GROUPS,DETAIL
EZD1254I GWM GROUP DETAIL
GWM LABEL        : GWM1
 GWM STATUS      : GWM_ACTIVE
 GWM TIMESTAMP   : 04/27/2005 12:32:01
 GWM IPADDR..PORT: 10.81.1.1..3860
 LOCAL IPADDR    : 10.81.4.4
 UUID            : UUID1
 UPDATE INTERVAL : 60
 LAST UPDATE     : 04/27/2005 01:05:03
 GROUP LABEL     : FTP_GROUP
  GROUP NAME     : FTP.ZONE2.MYCORP.COM
  GROUP TYPE     : SERVER
  DNS LABEL      : DNS2
  ZONE LABEL     : ZONE2
  MEMBER HOSTNAME:
   IPADDR..PORT  : 2001:0DB8:10::81:2:2..21
    PROTOCOL     : TCP
    AVAIL        : YES
    FLAGS        :
    UPDATE COUNT : 2
  MEMBER HOSTNAME: FTP3
   IPADDR..PORT  : 10.81.3.3..621
    PROTOCOL     : TCP
    AVAIL        : NO
    FLAGS        : NOTARGETSYS NOTARGETAPP
    UPDATE COUNT : 3
   IPADDR..PORT  : 2001:DB8:10::81:3:3..621
    PROTOCOL     : TCP
    AVAIL        : YES
    FLAGS        :
    UPDATE COUNT : 5
 GROUP LABEL     : HOST_GROUP
  GROUP NAME     : HOST7.ZONE7.MYCORP.COM
  GROUP TYPE     : HOST
  DNS LABEL      : DNS7
  ZONE LABEL     : ZONE7
  MEMBER HOSTNAME:
   IPADDR        : 10.81.7.7
    AVAIL        : YES
    FLAGS        :
    UPDATE COUNT : 1
  MEMBER HOSTNAME: HOST5V6
   IPADDR        : 2001:DB8:10::81:7:7
    AVAIL        : NO
    FLAGS        : NOTARGETSYS NOTARGETHOST
    UPDATE COUNT : 1
2 of 2 RECORDS DISPLAYED
```

GWM LABEL

The GWM label configured in the automated domain name registration application configuration file on the gwm statement.

GWM STATUS

The status of the GWM advising the automated domain name registration application. Possible values are:

CONNECTED

The automated domain name registration application is connected to the GWM specified.

CONVERGENCE_PENDING

The automated domain name registration application is waiting a fixed period of time for information about all configured groups to be returned from the GWM.

DISCONNECTED

The automated domain name registration application is not connected to the GWM.

GETWEIGHTS_RSP_PENDING

The automated domain name registration application is waiting for a SASP GetWeights response message from the GWM.

GWM_ACTIVE

The state of the GWM after it has exited the CONVERGENCE_PENDING state. This is the normal steady state of the GWM. When the GWM is in this state, all changes in the status of any configured group are received by the automated domain name registration application and forwarded to the appropriate name servers. The GWM remains in this state until there is a configuration change or until either the GWM or the automated domain name registration application is stopped.

PRE_REG_DEREGISTRATION_RSP_PENDING

The automated domain name registration application is waiting for a SASP DeRegistration response message from the GWM as a result of GWM communication initialization.

REGISTRATION RSP_PENDING

The automated domain name registration application is waiting for a SASP Registration response message from the GWM.

SETLBSTATE RSP_PENDING
The automated domain name registration application is waiting for a SASP SetLoadBalancerState response message from the GWM.

SHUTTING_DOWN
The automated domain name registration application is terminating.

GWM TIMESTAMP
The timestamp in UTC format specifying when the GWM reached the status indicated by GWM STATUS.

GWM IPADDR..PORT
The remote IP address and port of the GWM.

LOCAL IPADDR
The local IP address that the automated domain name registration application used to connect to the GWM.

UUID
The universally unique identifier that the automated domain name registration application used to connect to the GWM.

UPDATE INTERVAL
The GWM's update interval in seconds. See the appropriate GWM documentation for more information.

LAST UPDATE
The timestamp in UTC format specifying the most recent update (SASP SendWeights message) received from the GWM.

GROUP LABEL
The group label configured in the automated domain name registration application configuration file on the host_group statement or on the server_group statement.

GROUP NAME
The group name registered with the GWM. The group name is defined in the automated domain name registration application configuration file using the host_group_name parameter on the host_group statement or the server_group_name parameter on the server_group statement concatenated to the domain_suffix of the zone identified by the dns and zone parameters on the host_group statement or the server_group statement.

GROUP TYPE
The group type. Possible values are:

HOST
Indicates a host group.

SERVER
Indicates a server group.

DNS LABEL
The DNS label configured in the automated domain name registration application configuration file on the dns statement.

ZONE LABEL
The zone label configured in the automated domain name registration application configuration file using the zone parameter on the dns statement.

MEMBER HOSTNAME
The optional member *hostname* is defined in the automated domain name

registration application configuration file using the member host_name parameter on the host_group statement or the server_name parameter on the server_group statement.

IPADDR[..PORT]
The IP address and port on which the application can be reached. The port is not displayed when the GROUP TYPE value is HOST.

PROTOCOL
The protocol the application is using. The protocol value is either TCP or UDP. The protocol is not displayed when GROUP TYPE is HOST.

AVAIL
Indicates whether or not the member is available in the sysplex.

FLAGS
Indicates which flags are currently set. Flag values are:

NOTARGETAPP
The GWM found the IP address but did not find an available server application using the IP address, port, and protocol. When the GWM is the z/OS Load Balancing Advisor, this can indicate that the application member has been quiesced by the Agent.

NOTARGETHOST
The GWM found the IP address but the host is not available. When the GWM is the z/OS Load Balancing Advisor, this indicates that the system member has been quiesced by the Agent.

NOTARGETSYS
The GWM did not find the IP address.

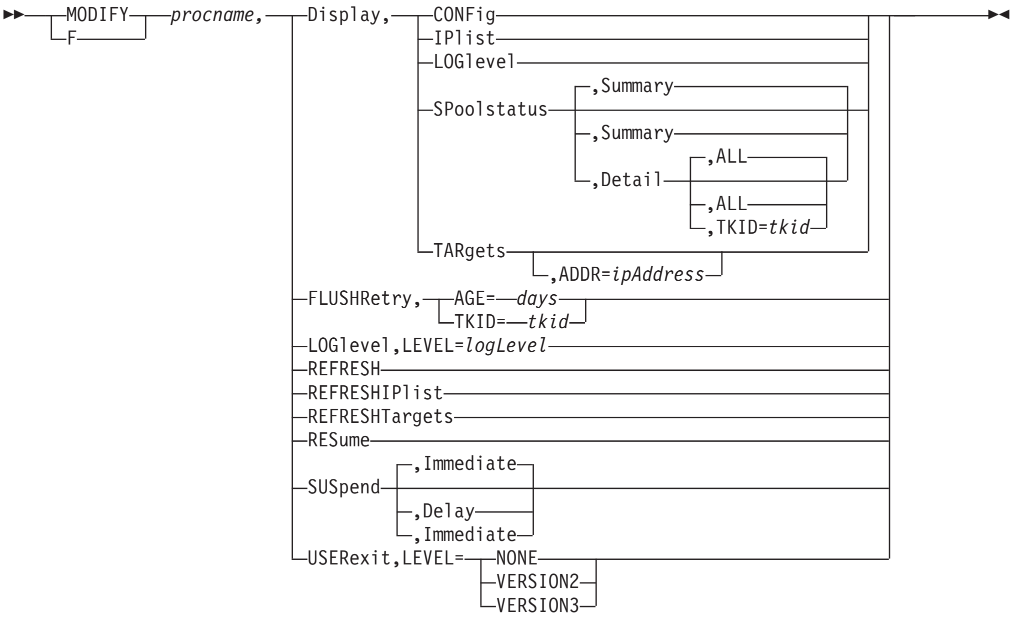
No flags are displayed when the AVAIL value is YES.

UPDATE COUNT
The number of times the availability of this IP address, which is associated with the preceding MEMBER HOSTNAME value, has changed.

MODIFY command: Communications Server SMTP application (CSSMTP)

Use the MODIFY command to control the Communications Server SMTP (CSSMTP) application from the operator console. For descriptions of terms that are used in this section, see the CSSMTP information in z/OS Communications Server: IP Configuration Guide.

Format



Parameters

procname
The member name of the cataloged procedure that is used to start the CSSMTP application.

Display,CONFig
Display the CSSMTP application configuration and global values that are used for processing mail.

Display,IPlist
Display all target server IP addresses and their preferences that are used by CSSMTP. A target server is the resolved or configured IP addresses from TargetServer statements. See the TargetServer statement information in z/OS Communications Server: IP Configuration Reference for details about how the target server addresses are obtained.

DISplay,LOGlevel
Display CSSMTP active log levels.

Display,SPoolstatusDisplay,SPoolstatus,Summary
Display summary information for all tasks that are processing spool files for CSSMTP. You can use this display to determine the number of mail messages that are pending or you can use it on the long-retry queue for each spool file that is being processed. The summary option is the default for the MODIFY DISPLAY,SPOOLSTATUS command.

Tip: Use this command to obtain the task ID to use on other modify commands that use task ID values as options.

Display,SPoolstatus,DetailDisplay,SPoolstatus,Detail,ALL
Display detailed information for all tasks that are not idle. The ALL option is the default for the MODIFY DISPLAY,SPOOLSTATUS,DETAIL command.

Display,SPoolstatus,Detail,TKID=*tkid*
Display detailed information for this specific task that processes spool files for CSSMTP.

Tip: You can use the MODIFY DISPLAY,SPOOLSTATUS command to obtain a valid TKID value.

Display,TARgets[,ADDR=*ipAddress*]

Display global and specific information about target servers. If the ADDR parameter is not specified, all configured target servers are displayed. If the ADDR parameter is specified, then the IP address value must match the IP address of an existing target server that is in use by CSSMTP.

Tip: You can use the MODIFY DISPLAY IPLIST command to obtain IP addresses for the list of target servers that are being used by CSSMTP.

FLUSHRetry,TKID=*tkid*

Initiate a request for the CSSMTP application to remove mail messages from the long retry queue, and send those mail messages to the list of target servers. If CSSMTP cannot send a mail message, that mail message becomes subject to long-term retry processing; if any mail message is not defined, it becomes an undeliverable mail message. For more information about undeliverable mail, see z/OS Communications Server: IP Configuration Guide.

A nonzero TKID (task ID) value requests that only the mail messages for the specified TKID value is flushed. A TKID value 0 requests that all mail messages in the long-retry queue be flushed.

Tip: You can use the MODIFY DISPLAY,SPOOLSTATUS command to obtain a valid TKID value.

FLUSHRetry,AGE=*days*

Initiate a request for the CSSMTP application to remove the mail messages that are older than *days* days from the extended retry queue, and send those mail messages to the list of target servers. If CSSMTP cannot send a mail message, that message becomes an undeliverable mail message. To make this command effective, the target servers must be available. For details, see the information about extended retry mail in z/OS Communications Server: IP Configuration Reference and the information about undeliverable mail in z/OS Communications Server: IP Configuration Guide.

A days value of 0 specifies that all messages in the extended retry list are to be processed.

Tip: You can use the command to monitor the number of the mail messages in the extended retry list and the state of the target servers used by CSSMTP.

LOGlevel,LEVEL=*logLevel*

Change the CSSMTP application log level. The *logLevel* value specifies the log level. If a *logLevel* value is not specified, then the current log level remains in effect. See the LogLevel statement information in z/OS Communications Server: IP Configuration Reference for details about defining the CSSMTP application log level.

REFRESH

Initiate a dynamic reconfiguration using the configuration file that is defined at initialization. If a configuration error is detected during a dynamic refresh, the entire refresh is rejected, the error message is written to the log and console, and the CSSMTP application continues to run with the old configuration values.

Results:

- While the new configuration file is being processed, the existing log level is used, regardless of how it was set (using the last configuration file or with the MODIFY LOGLEVEL command). After the new configuration file has been successfully processed, the value that is specified on the LogLevel statement of the new configuration file takes effect. If the LogLevel statement is not specified in the new configuration file, the log level defaults to level 7 (ERROR, WARNING, and EVENT). If the new configuration file contains errors that cause it to be rejected, the log level that was in effect prior to the dynamic reconfiguration is used.

- If an update to the ExtWrtName statement is detected during a dynamic refresh, then the CSSMTP application continues to run with the old external writer name and a warning message is written to the log and console.
- If an update to the Translate statement is detected during a dynamic refresh, then the CSSMTP application continues to run with the old *translate* value and a warning message is written to the log and console.
- If an update to the ChkPointSizeLimit statement is detected during a dynamic refresh, then the CSSMTP application continues to run with the old ChkPointSizeLimit value and a warning message is written to the log and console.
- While the new configuration file is being processed, the existing UserExit value is used, regardless of how it was set (using the last configuration file or with the MODIFY USEREXIT command). After the new configuration file has been successfully processed, the value that is specified on the UserExit statement of the new configuration file takes effect when the next JES spool file is opened. If the new configuration file contains errors that cause it to be rejected, the UserExit value that was in effect prior to the dynamic reconfiguration is used.
- An update to the TargetServer statement can force CSSMTP to stop and restart connections on the affected IP addresses. If CSSMTP is in the process of sending a mail message on the affected IP address, the mail message is retried at another IP address or placed in the long retry queue. For more information about the TargetServer statement, see the TargetServer statement information in z/OS Communications Server: IP Configuration Reference.

REFRESHIPlist

Initiates a dynamic DNS refresh of the target that is identified by the configured TargetName or TargetMx parameter value. This parameter does not cause the configuration file to be reprocessed.

Result: If a TargetServer statement has TargetName or TargetMx parameters configured, new IP addresses might be resolved. If the IP address list is changed, CSSMTP might be forced to stop and restart connections on the affected IP addresses. If CSSMTP is in the process of sending a mail message on the affected IP address, the mail message is retried at another IP address or placed in the long retry queue.

REFRESHTargets

Reinitiates a connection to all target servers. The CSSMTP application can learn about any capability changes from the target servers.

Tip: This command causes all active connections to all target servers to be stopped and restarted; therefore, use this command only when there is a change in the network topology and no work is being done by the CSSMTP application because the command interrupts all active connections.

Result: If you issue this command while CSSMTP is in the process of sending a mail message on the connection, the mail message is retried at another IP address or placed in the long retry queue.

RESume
Resumes processing of any JES spool files whose processing was suspended with the MODIFY SUSPEND operator command.

SUSpendSUSpend,Immediate
Suspends the reading of mail messages immediately for all active spool files. To resume this processing, issue the MODIFY RESUME operator command. The IMMediate option is the default for the MODIFY SUSPEND command.

SUSpend,Delay
Suspends the reading of any new spool files immediately but completes reading any spool files that are already in process. To resume reading of spool files, issue the MODIFY RESUME operator command.

USERexit,LEVEL=*userExitValue*
Change the CSSMTP application user exit value. The USERexit keyword can be set to NONE, VERSION2, or VERSION3. If a *userExitValue* parameter value is not specified, then the current user exit value remains in effect. See the USEREXIT statement information in z/OS Communications Server: IP Configuration Reference for details about how to define the CSSMTP application user exit value.

Result: The user exit value does not change until the next JES spool file is opened.

Examples

Example 1: The MODIFY DISPLAY,LOGLEVEL command displays the current logging level that is being used by CSSMTP.

```
F CSSMTP,DISPLAY,LOGLEVEL
   EZD1828I CSSMTP DISPLAY LOGLEVEL = 15
```

Example 2: The MODIFY DISPLAY,CONFIG command displays the current configuration that is being used by CSSMTP.

```
F CSSMTP,DISPLAY,CONFIG
  EZD1829I CSSMTP CONFIGURATION:
   CONFIGFILENAME       : /U/USER1/CSSMTP/CSSMTP.CONF
   LOGFILENAME          : /U/USER1/CSSMTP/CSSMTP.LOG0707S1
   CHKPOINTFILENAME     : 'USER1.CSSMTP.CHKPOINT'
   PID                  : 67108874
   LOGLEVEL             : 255         USEREXIT        : NONE
   CHKPOINTSIZELIMIT    : 64000       CHKPOINT        : WARM
   CONFIG CODEPAGE      : IBM-1047
   TRANSLATE            : IBM-1047
   START OPTION TCPNAME : N/A         IPV6 ENABLED    : YES
   EXTWRTNAME           : CSSMTP      HOST NAME       : VIC142
   DOMAIN NAME          : RALEIGH.IBM.COM
   HEADER               :
    DATE                : YES
    USERINFO            : YES
   JESJOBSIZE           : 0           JESMSGSIZE      : 0
   JESSYNTAXERRLIMIT    : 5
   BADSPOOLDISP         : HOLD        REPORT          : SYSOUT
   UNDELIVERABLE:
    RETURNTOMAILFROM    : YES         DEADLETTERACTION: STORE
   DEADLETTERDIRECTORY : /var/cssmtp/myDir/
   RETRYLIMIT:
    COUNT               : 5           INTERVAL        : 1
   EXTENDEDRETRY        : ACTIVE
    AGE                 : 100         INTERVAL        : 30
    MAILDIRECTORY       : /var/cssmtp/CSSMTP/mail/
   SMF119:
     CONFIG            : YES        CONNECT         : YES
     MAIL              : YES        SPOOL           : YES
     STATS             : YES

   TARGETSERVER:
    TARGETNAME         : D03AV01.BOULDER.IBM.COM
     CONNECTPORT       : 25         CONNECTLIMIT    : 5
     MAXMSGSENT        : 1000       MESSAGESIZE     : 524288
     SECURE            : NO
    TARGETIP           : 9.100.1.2
     CONNECTPORT       : 25         CONNECTLIMIT    : 5
     MAXMSGSENT        : 1000       MESSAGESIZE     : 524288
     SECURE            : NO
   TIMEOUT:
     ANYCMD            : 300        CONNECTRETRY    : 120
     DATABLOCK         : 180        DATAINIT        : 120
     DATATERM          : 600        INITIALMSG      : 300
     MAILCMD           : 300        RPCTCMD         : 300
   MAILADMINISTRATOR   : USER1@US.IBM.COM
                          USER2@US.IBM.COM
                          USER3@US.IBM.COM
                          USER4@US.IBM.COM
```

For definitions of statements and parameters that are obtained from configuration file, see the CSSMTP information in z/OS Communications Server: IP Configuration Reference.

CONFIGFILENAME
The configuration name from the CONFIG DD statement in the started procedure.

LOGFILENAME
The configured log file name from LOGFILE DD statement in the started procedure.

PID
The process ID.

LOGLEVEL
The logging level.

CHKPOINT
Indicates whether checkpointing is active.

WARM
Checkpointing was initiated using the CHKPOINT DD statement.

COLD
Checkpointing was initiated using the **-f** start option.

NONE
There was no CHKPOINT DD statement.

CONFIG CODEPAGE
The code page value specified on the CSSMTP_CODEPAGE_CONFIG statement or the default value.

TRANSLATE
The code page value that is configured on the Translate statement.

CHKPOINT FILENAME
The name of the configured checkpoint file, if a CHKPOINT DD statement is configured in the started procedure.

START OPTION TCPNAME
The TCP name that is passed on the **-p** start option or the value N/A if the **-p** start option is not used.

IPV6 ENABLED
Indicates whether IPV6 is supported.

EXTENDEDRETRY
Indicates whether extended retry processing is ACTIVE or INACTIVE.

The remaining values that are displayed are the values for the matching statement or parameter from the configuration file.

Example 3: The MODIFY DISPLAY,IPLIST command displays the IP address of the configured target servers from the TargetServer statement and TargetIp parameter, or it displays the resolved target server addresses from the TargetServer statement and TargetName parameter, that are being used by CSSMTP.

```
F CSSMTP,DISPLAY,IPLIST
EZD1830I CSSMTP IPLIST:
   TARGETIP           : 9.100.1.5
    CONNECTPORT       : 25         CONNECTLIMIT   : 5
    MAXMSGSENT        : 1000       MESSAGESIZE    : 524288
    SECURE            : NO
   TARGETIP           : 9.56.231.69
    CONFIG TARGETNAME : RALVMS
    CONNECTPORT       : 25         CONNECTLIMIT   : 2
    MAXMSGSENT        : 2000       MESSAGESIZE    : 524288
    SECURE            : YES
   TARGETIP           : 9.200.1.6
    CONNECTPORT       : 25         CONNECTLIMIT   : 5
    MAXMSGSENT        : 1000       MESSAGESIZE    : 524288
    SECURE            : NO
```

For the definitions of statements and parameters that are obtained from the configuration file, see the CSSMTP information in z/OS Communications Server: IP Configuration Reference.

TARGETIP
The IP address of the target server.

CONFIG TARGETNAME
The name that is used to resolve this target server address for a resolver A or AAAA query.

The remaining values that are displayed are the same values that are specified on the matching statement or parameter in the configuration file.

Example 4: The MODIFY DISPLAY,IPLIST command displays the resolved target servers from the TargetServer statement and TargetMx parameter that are being used by CSSMTP.

```
F CSSMTP,DISPLAY,IPLIST
EZD1830I USER1408 IPLIST:
   TARGETIP           : 9.56.231.69
    CONFIG TARGETMX   : mxName
    PREFERENCE        : 1
    CONNECTPORT       : 25         CONNECTLIMIT   : 2
    MAXMSGSENT        : 2000       MESSAGESIZE    : 524288
    SECURE            : YES
   TARGETIP           : 9.56.200.55
    CONFIG TARGETMX   : mxName
```

```
PREFERENCE        : 1
CONNECTPORT       : 25         CONNECTLIMIT   : 2
MAXMSGSENT        : 2000       MESSAGESIZE    : 524288
SECURE            : NO
```

For the definition of statements and parameters that are obtained from configuration file, see the CSSMTP information in z/OS Communications Server: IP Configuration Reference.

TARGETIP

This is the IP address of the target server.

CONFIG TARGETMX

The name that is used to resolve this target server address for resolver MX query.

The remaining values that are displayed are the values for the matching statement or parameter from the configuration file.

Example 5: The MODIFY DISPLAY,SPOOLSTATUS command displays the summary information for all tasks that are processing JES spool files.

```
F CSSMTP,DISPLAY,SPOOLSTATUS
   EZD1832I CSSMTP SPOOLSTATUS:
  TKID JOBNAME  STATE  PEND   LRT     TKID JOBNAME  STATE  PEND   LRT
 W 002 JOBNM25  ACTVE    15     5    D 003 JOBNM132 READ      5     0
 W 004 JOBNM45  ACTVE    10    10    D 005 JOBNM232 READ     10     0
 W 006 JOBNM48  ACTVE    20     0    D 007 JOBNM332 IDLE      0     0
 W 008 JOBNM50  ACTVE    20     5    D 009 JOBNM432 IDLE      0     0
 W 010 JOBNM60  ACTVE    10     0    D 011 JOBNM532 IDLE      0     0
 W 012 JOBNM80  ACTVE     0     0    D 013 JOB00632 IDLE      0     0
 W 014 JOBNM90  ACTVE    10    10    D 015 JOB00732 IDLE      0     0
 W 016 JOBNM190 ACTVE    20     0    D 017 JOB00832 IDLE      0     0
 W 018 JOBNM150 ACTVE    20     5    D 019 JOB00932 IDLE      0     0
 W 020 JOBNM160 ACTVE    10     0    D 021 JOB01132 IDLE      0     0
```

For definitions of terms that relate to this information, see the CSSMTP common terms information in z/OS Communications Server: IP Configuration Guide.

W A writer JES task, if the JES spool file was generated by the IEBGENER utility.

D A dest JES task, if the JES spool file was generated by the SMTPNOTE command or by the TSO Transmit (XMIT) command.

TKID

The task ID, which can be used to identify a specific task.

Tip: You can use the task ID in the MODIFY FLUSHRetry command and the MODIFY DISPLAY,SPOOLSTATUS,DETAIL,TKID=*tkid* command.

JOBNAME

The JES job name for this task. If the task is in the IDLE state, this is the name of the previous job.

STATE

This parameter can have one of the following values:

WAITS

The task is waiting because virtual storage is constrained.

READ

The task is reading a spool file.

IDLE

The task is waiting for a JES spool file to process.

  ACTVE
    The task is actively waiting for all mail in the spool file to be processed.

  WAIT
    The task is waiting because no target server is active to receive mail.

  SUPND
    The task was suspended by the MODIFY SUSPEND command.

PEND
  The number of mail messages that are waiting to be sent to target server.

LRT
  The number of mail messages that are currently queued in the long-retry queue.

The MODIFY DISPLAY SPOOLSTATUS,DETAIL command displays detailed information for all tasks that are not in the IDLE state that are processing JES spool files.

```
F CSSMTP,DISPLAY,SPOOLSTATUS,DETAIL
EZD1833I CSSMTP SPOOLSTATUS:
 TASK ACTVE          : WRITER     TKID         : 2
  JOBNAME            : JOBNM25    JOBID        : STC00055
  PEND               : 15         LRT          : 5
  MAIL READ          : 0          UNDELIVERABLE: 0
 TASK ACTVE          : WRITER     TKID         : 4
  JOBNAME            : JOBNM45    JOBID        : STC00060
  PEND               : 10         LRT          : 10
  MAIL READ          : 10         UNDELIVERABLE: 0

... <active task TKID=006,008,010,012,014,016,018,020 not shown>
 TASK READ           : DEST       TKID         : 3
  JOBNAME            : JOBNM132   JOBID        : STC00055
  PEND               : 5          LRT          : 0
  MAIL READ          : 5          UNDELIVERABLE: 20
 TASK READ           : DEST       TKID         : 5
  JOBNAME            : JOBNM232   JOBID        : STC00255
  PEND               : 10         LRT          : 0
  MAIL READ          : 5          UNDELIVERABLE: 20
```

The MODIFY DISPLAY SPOOLSTATUS,DETAIL command displays detailed information for all tasks that are not in the IDLE state that are processing JES spool files; in this example all state tasks are idle.

```
F CSSMTP,Display,Spoolstatus,detail
EZD1833I CSSMTP SPOOLSTATUS:
No non-idle TKIDs to display
```

The MODIFY DISPLAY SPOOLSTATUS,DETAIL,TKID=11 command displays detailed information for a specific TKID 11 task that is processing a spool file for the CSSMTP application.

```
F CSSMTP,DISPLAY,SPOOLSTATUS,DETAIL,TKID=11
EZD1833I CSSMTP SPOOLSTATUS:
 TASK ACTVE          : WRITER     TKID         : 11
  JOBNAME            : JOBNM532   JOBID        : STC00532
  PEND               : 0          LRT          : 0
  MAIL READ          : 25         UNDELIVERABLE: 0
```

The MODIFY DISPLAY SPOOLSTATUS,DETAIL,TKID=3 command displays detailed information for a specific TKID 3 task that is currently idle.

```
F CSSMTP,DISPLAY,SPOOLSTATUS,DETAIL,TKID=2
EZD1833I CSSMTP SPOOLSTATUS:
 TASK IDLE           : WRITER      TKID            : 2
```

```
JOBNAME          : JOB00032   JOBID          : STC00055
PEND             : 0          LRT            : 0
MAIL READ        : 2          UNDELIVERABLE  : 1
```

TASK
The TASK parameter has the following fields:

state
Possible values are:

WAITS
The task is waiting because virtual storage is constrained.

READ
The task is reading a spool file.

IDLE
The task is waiting for a JES spool file to process.

ACTVE
The task is actively waiting for all mail in the spool file to be processed.

WAIT
The task is waiting because no target server is active to receive mail.

SUPND
The task was suspended by the MODIFY SUSPEND command.

type task
Possible **type task** values are:

WRITER
This is a writer JES task if the JES spool file was generated by the IEBGENER utility.

DEST
This is a DEST JES task if the JES spool file was generated by the SMTPNOTE command or by the TSO Transmit (XMIT) command.

TKID
The task ID, which can be used to identify a specific task.

Tips:

- You can use the task ID in the MODIFY FLUSHRetry command and the MODIFY DISPLAY,SPOOLSTATUS,DETAIL,TKID=*tkid* command.
- You can use the task ID to identify log information that is in the log file. If the *tkid* value is 2, that TKID value is represented in the following example by the value :002.

```
08/01 07:10:20 CSSMTP DEBUG  :002:mlJESThread:Message(0): ...
```

JOBNAME
The JES job name for this task. If the task is in the IDLE state, then this is the previous job name.

JOBID
This is the JES job ID for this task. If the task is in the IDLE state, then this is the previous job ID.

PEND
The number of mail messages that are waiting to be sent to a target server. If the task is in the IDLE state, then this value is always 0.

LRT
The number of mail messages that are currently in the long-retry queue. If the task is in the IDLE state, then this value is always 0.

MAIL READ
The total number of mail messages that have been read for the job. If the task is in the IDLE state, then this is the mail for the previous job.

UNDELIVERABLE
The total number of undeliverable mail messages for this job name. If the task is in IDLE state, then this value is the undeliverable count of the previous job.

Example 6: The MODIFY DISPLAY,TARGETS command displays the global and specific information that is related to sending email to target servers. For definitions of statements and parameters that are obtained from the configuration file, see the CSSMTP information in z/OS Communications Server: IP Configuration Reference.

```
F CSSMTP,DIS,TARGETS
 EZD1831I CSSMTP TARGETS:
  GLOBAL INFORMATION:
   MAIL SENT : 0                     TOTAL RETRY  : 0
   DEADLETTER: 0                     CURRENT RETRY: 0
   UNDELIVER : 0
  EXTENDED RETRY:
   CURRENT   : 0                     TOTAL        : 0
  TARGET SERVER 1.1.1.1
   STATE     : ACTIVE
   ESMTP     : YES                   MESSAGE SIZE : 20000000
   STARTTLS  : NO                    MAIL ATTEMPTS: 24
   MAIL SENT : 24                    CONNECT FAIL : 0
  TARGET SERVER ::6
   STATE     : ACTIVE
   ESMTP     : YES                   MESSAGE SIZE : 20000000
   STARTTLS  : YES                   MAIL ATTEMPTS: 30
   MAIL SENT : 30                    CONNECT FAIL : 0
```

The following global target server information for this application is displayed:

MAIL SENT
Count of all mail messages that were processed successfully after all mail messages were sent.

TOTAL RETRY
Cumulative count of mail messages that have been in the long-retry state.

DEADLETTER
Cumulative count of all mail messages that were classified as dead letters.

CURRENT RETRY
Count of mail messages that are currently in the long-retry queue.

UNDELIVER
Count of all mail messages that were undeliverable.

EXTENDED RETRY

CURRENT
Count of mail messages currently in the extended retry directory while extended retry is active.

TOTAL
Cumulative count of mail messages that have been in the extended retry directory.

The following fields are displayed for each target server:

STATE
State of the target server.

ACTIVE
The target server is active.

NOT USABLE
This target server is not usable. For example, CSSMTP has lost connectivity to this target server.

UNKNOWN
This target server is new and its capabilities have not been learned at this time.

ESMTP
Type of target server. If the ESMTP value is YES, the target server type is ESMTP. If the ESMTP value is NO, the target server type is SMTP.

MESSAGE SIZE
The actual message size.

- For ESMTP this value was obtained from the SIZE extension when the connection was made. The value 0 indicates that there is no predefined message size limit.
- For SMTP, this value is the value that is configured for the MessageSize parameter of the TargetServer statement. For details, see the TargetServer statement information in z/OS Communications Server: IP Configuration Reference

STARTTLS
Indicates that the target server has acknowledged that it is capable of establishing secure connections.

- For ESMTP see the information that describes using Transport Layer Security TLS in z/OS Communications Server: IP Configuration Guide for details.
- For SMTP this is always set to NO.

MAIL ATTEMPTS
The total number of mail messages that CSSMTP has sent or has attempted to send to the target server.

Tip: A single mail message operation can be attempted on multiple target servers.

MAIL SENT
The number of mail messages that were sent successfully for this target server.

Tip: A single mail message that contains multiple recipients can be sent successfully on multiple target servers.

CONNECT FAIL
Count of the number of SMTP connections that the application was unable to establish when it attempted to send mail to a target server.

Example 7: The MODIFY DISPLAY,TARGETS,ADDR=*x.x.x.x* command displays the global and specific information related to sending e-mail to a specific target server.

```
F CSSMTP,DIS,TARGETS,ADDR=1.1.1.1
 EZD1831I CSSMTP TARGETS:
  GLOBAL INFORMATION:
   MAIL SENT : 24                   LONG RETRY   : 0
   DEADLETTER: 0                    UNDELIVERABLE: 0
  TARGET SERVER 1.1.1.1
```

```
STATE     : ACTIVE
ESMTP     : YES                MESSAGE SIZE : 20000000
SECURE    : NO                 MAIL ATTEMPTS: 24
MAIL SENT : 24                 CONNECT FAIL : 0
```

Example 8: The MODIFY LOGLEVEL,LEVEL=15 command requests the CSSMTP application to change the logging level to 15. The EZD1809I message indicates that the logging level was updated successfully.

```
F CSSMTP,LOGLEVEL,LEVEL=15
   EZD1809I CSSMTP1 MODIFY LOGLEVEL COMMAND COMPLETED : UPDATED
```

Example 9: The MODIFY REFRESH command requests that the CSSMTP application reprocess the configuration file. The following messages indicate that the configuration was updated successfully with no errors.

```
F CSSMTP,REFRESH
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1840I CSSMTP UPDATED CONFIGURATION
   EZD1846I CSSMTP UPDATED TARGET SERVERS
   EZD1848I CSSMTP MODIFY REFRESH COMMAND COMPLETED
```

Example 10: The MODIFY REFRESHIPLIST command requests that the CSSMTP application perform a dynamic DNS refresh of the TargetName or TargetMx field. The following messages indicate that the target server addresses were successfully updated.

```
F CSSMTP,REFRESHIPLIST
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1845I CSSMTP UPDATED TARGET SERVERS
   EZD1842I CSSMTP MODIFY REFRESHIPLIST COMMAND COMPLETED
```

Example 11: The MODIFY FLUSHRETRY,TKID=0 command initiates a request to move all mail that is in the long-retry queue to the send queue.

The following messages indicate that all mail messages have been moved from the long-retry queue to the send queue.

```
F CSSMTP,FLUSHRETRY,TKID=0
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1810I CSSMTP MODIFY FLUSHRETRY,TKID=0 COMMAND COMPLETED
```

Example 12: The MODIFY FLUSHRETRY command initiates a request to move all mail for TKID 2 from the long-retry queue to the send queue.

The following messages indicate that all mail messages for TKID 2 have been moved from the long-retry queue to the send queue.

```
F CSSMTP,FLUSHRETRY,TKID=2
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1810I CSSMTP MODIFY FLUSHRETRY,TKID=2 COMMAND COMPLETED
```

Example 13: The MODIFY REFRESHTARGETS command reinitiates a connection to all target servers.

The following messages indicate that the CSSMTP application completed this request.

```
F CSSMTP,REFRESHTARGETS
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1821I CSSMTP ABLE TO USE TARGET SERVER 9.1.1.1
```

Example 14: The MODIFY RESUME command resumes processing of any JES spool files when processing was suspended using the MODIFY SUSPEND operator command.

The following messages indicate that the CSSMTP application will start processing any JES spool files.

```
F CSSMTP,RESUME
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1814I CSSMTP MODIFY RESUME COMMAND COMPLETED
```

Example 15: The MODIFY SUSPEND command suspends the reading of mail messages immediately for all JES spool files.

The following messages indicate that the CSSMTP application has suspended the processing of mail messages for all JES spool files.

```
F CSSMTP,SUSPEND
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1822I CSSMTP MODIFY SUSPEND IMMEDIATE COMMAND COMPLETED
```

Example 16: The MODIFY FLUSHRETRY,AGE=0 command initiates a request to move all the mails that are in the extended retry queue to the send queue.

The following messages indicate that all mail messages have been moved from the extended retry queue to the send queue.

```
F CSSMTP,FLUSHRETRY,AGE=0
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1823I CSSMTP MODIFY FLUSHRETRY,AGE=0 COMMAND COMPLETED
```

Example 17: The MODIFY FLUSHRETRY,AGE=2 command initiates a request to move all mail messages that have been in the extended retry queue for more than two days from the extended retry queue to the send queue.

The following messages indicate that all the extended retry mail messages that are older than two days have been moved from the extended retry queue to the send queue.

```
F CSSMTP,FLUSHRETRY,AGE=2
   EZD1834I CSSMTP MODIFY COMMAND ACCEPTED
   EZD1823I CSSMTP MODIFY FLUSHRETRY,AGE=2 COMMAND COMPLETED
```

MODIFY command: DCAS

Use the MODIFY command to control tracing after initialization is complete.

Format

```
►►──┬─MODIFY─┬──jobname_,_─,─DEBUG_=debug_level──────────────────────────────►◄
    └─F──────┘
```

Parameters

DEBUG
debug_level

Controls digital certificate access server (DCAS) general tracing. The *debug_level* value must be one of the following values:

0 Disables logging.

1 Specifies log error and warning messages.

2 Specifies log error, warning, and informational messages.

3 Specifies log error, warning, informational, and debug messages.

MODIFY command: Defense Manager daemon

Use the MODIFY command to control the Defense Manager (DM) functions from the operator console.

Format

```
├──┬─MODIFY─┬──procname,DISPLAY──────────────────────────────────────────────┤
   └─F──────┘
```

```
├──┬─MODIFY─┬──procname,REFRESH──┬───────────────────────┬────────────────────┤
   └─F──────┘                     ├─,FILE='filename'──────┤
                                  └─,FILE=//'filename'────┘
```

```
├──┬─MODIFY─┬──procname,FORCE_INACTIVE,stackname───────────────────────────────┤
   └─F──────┘
```

Parameters

procname
The member name of the cataloged procedure that is used to start the Defense Manager daemon (DMD).

DISPLAY
Displays configuration values that are currently being used by the DMD.

REFRESH
Indicates that the DMD configuration file should be reread. The file is treated as a complete replacement, so it must contain all necessary DMD configuration information. You cannot update all DMD parameters using this command. See the description for the parameters in the configuration file to find out which ones can be dynamically changed. You must include parameters that cannot be dynamically changed in the REFRESH configuration file if the daemon was started with a value for the parameter that was not the default value. See the Defense Manager daemon information in the z/OS Communications Server: IP Configuration Reference.

FILE
Indicates the name and location of the DMD configuration file that is to be read. The file is treated as a complete replacement so it must contain all necessary DMD configuration information. The file name must be a fully qualified z/OS UNIX file name or MVS data set name. A z/OS UNIX file name must be enclosed by single quotation marks ('). MVS data set names must begin with two forward slashes (//) and the data set name must be enclosed by single quotation marks ('). This option is valid only when specified with REFRESH.

FORCE_INACTIVE ,*stackname*
Forces the TCP/IP stack named *stackname* to become inactive with respect to defensive filters. All defensive filters for the stack are removed from DMD

persistent memory and also from the stack itself. No additional defensive filters are added to the stack while it is in inactive mode. The stack does not have to be configured in the DMD configuration file in order for the FORCE_INACTIVE option to operate. If the stack is active and IP security is enabled, then any defensive filters in the stack are removed regardless of the DMD configuration status of the stack. Changes to the mode of the stack persist until the next time the MODIFY *procname*,REFRESH command is successfully issued. See the Defense Manager daemon information in the z/OS Communications Server: IP Configuration Reference.

Examples

The following example displays the command and current configuration values.

```
f dmd,display
EZD1733I DISPLAY DMD CONFIGURATION
Defense Manager Configuration Settings
  SyslogLevel             = 7
  DefensiveFilterDirectory = /var/dm/filters
  DM Config for TCP/IP stack TCPCS
    Mode            = Simulate
    MaxLifetime     = 1440
    DefaultLogLimit = 100
    Exclude         192.168.1.3
    Exclude         192.168.1.10
```

The following example is the command and output used to forcibly deactivate a stack with respect to the Defense Manager daemon (DMD).

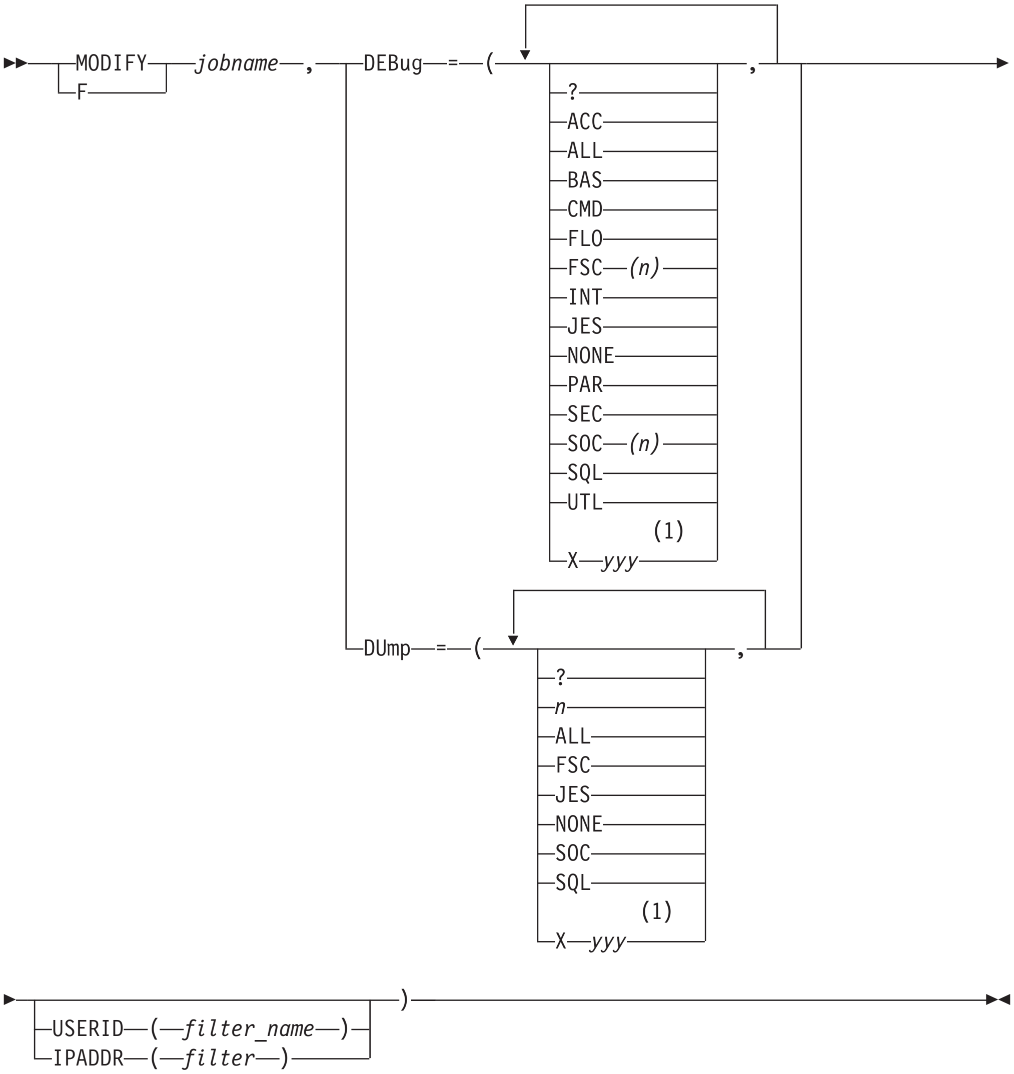
```
F DMD,FORCE_INACTIVE,TCPCS
EZD1643I THE DEFENSIVE FILTER MODE FOR STACK TCPCS WAS SUCCESSFULLY
        FORCED TO INACTIVE
```

MODIFY command: FTP

Use the MODIFY command to start and stop tracing after initialization is complete. The MODIFY command for z/OS FTP has two keywords: one for general tracing (DEBug) and one for extended tracing (DUMP).

Only FTP sessions established after trace is active can be traced. When tracing is stopped, sessions currently connected to the server will continue to be traced; new FTP sessions will not be traced.

Format



Notes:

1 Prepend any option *yyy* with X to turn off that trace.

Parameters

DEBug

Subcommand to begin general tracing. General tracing includes the following options:

? Displays the status of the general traces.

Note: The status of the trace is displayed as a response to all uses of the MODIFY DEBug command. The ? allows you to obtain the status without making a change.

ACC

Displays the details of the login process.

ALL

Sets all of the trace points.

Note: Both the FSC and the SOC trace are set to level 1 when the ALL parameter is processed.

BAS
Sets a select group of traces that offer the best overall details without the intense tracing of some of the traces. This is equivalent to:

```
MODIFY jobname,DEBUG=(CMD,INT,FSC,SOC)
```

CMD
Shows each command and the parsing of the parameters for the command.

FLO
Shows the flow of control within FTP. It is useful to show which services of FTP are used for an FTP request.

FSC*(n)*
Shows details of the processing of the file services commands APPE, STOR, STOU, RETR, DELE, RNFR, and RNTO. This trace can be very intense and therefore it allows you to specify levels of granularity for the trace points. The level 1 tracing that is specified by entering FSC or FSC(1) is the level that is normally used unless more data is requested by TCP/IP service group. The variable *n* can be a number in the range 1 – 8.

INT
Shows the details of the initialization and termination of the FTP session.

JES
Shows details of the processing for JES requests, such as when SITE FILETYPE=JES is in effect.

NONE
Turn off all of the traces.

PAR
Shows details of the FTP command parser. It is useful for debugging problems in the handling of the command parameters.

SEC
The SEC trace shows the processing of security functions such as TLS and GSSAPI negotiations.

SOC*(n)*
Shows details of the processing during the setup of the interface between the FTP application and the network as well as details of the actual amounts of data that is processed. This trace can be very intense and therefore it allows you to specify levels of granularity for the trace points. The level 1 tracing that is specified by entering SOC or SOC(1) is the level normally used unless more data is requested by the TCP/IP service group. The variable *n* can be a number in the range 1 – 8.

SQL
Shows details of the processing for SQL requests, such as when SITE FILETYPE=SQL is in effect.

UTL
Shows the processing of utility functions such as CD and SITE.

X*yyy*
Turns off an active option, where *yyy* is the option. For example: XUTL turns off the UTL option.

DUMp

Subcommand to begin extended tracing. Extended tracing includes the following options:

? Displays the status of the extended traces.

n Specifies the number of a specific extended trace point that is to be activated in the FTP code. The number has a range of 1 – 99.

ALL

Activates all of the trace points.

FSC

Activates all of the extended trace points in the file services code. The numbers activated are 20 – 49.

JES

Activates all of the extended trace points in the JES services code. The numbers activated are 60–69.

NONE

Turns off all extended traces.

SOC

Activates all of the extended trace points in the network services code. The numbers activated are 50–59.

SQL

Activates all of the extended trace points in the SQL services code. The numbers activated are 70–79.

X*yyy*

Turns off an active option, where *yyy* is the option. For example: XUTL turns off the UTL option.

USERID(*filter_name***)**

Filter the trace for user IDs matching the *filter_name* pattern.

If the user ID matches the filter at the time the client logs in, then tracing options are set to the current value of the options. Otherwise, there are no tracing options set. The client might use the SITE command to set options after login if the initial ones are not appropriate. An example for the USERID filter is `MODIFY jobname,DUMP=(21,USERID(USER33))` which activates the dumpID 21 trace for a user if the user ID is `USER33`.

IPADDR(*filter***)**

Filter the trace for IP addresses matching the *filter* pattern.

If the IP address matches the filter at the time the client connects, then tracing options will be set to the current value of the options. Otherwise, no tracing options will be set. The client might use the SITE command to set options after connect if the initial ones are not appropriate. An example of the IPADDR filter is `MODIFY jobname,DEBUG=(JES,IPADDR(9.67.113.57))` which will activate the JES trace for a client whose IP address is `9.67.113.57`. Specify the filter address in dotted decimal format if the IP address is an IPv4 address. Indicate submasking by using a slash followed by a dotted decimal submask. For example, `192.48.32/255.255.255.0` will allow addresses from `192.48.32.00` to `192.48.32.255`.

Specify the filter address for an IPv6 address as *x:x:x:x:x:x:x:x*, where the *x*s are the hexadecimal values of the eight 16-bit pieces of the address. Alternate notations described in RFC 2373 (IP Version 6 Addressing Architecture) are acceptable.

For example,

```
MODIFY jobname,DEBUG=(JES,IPADDR(FEDC:BA98:7654:3210:FEDC:BA98:7654:3210)
MODIFY jobname,DUMP=(FSC,IPADDR(::1))
```

Indicate IPv6 network prefixing using a slash followed by the number of prefix bits. For example, use `12AB:0:0:CD30::/60` to indicate the prefix `12AB00000000CD3` (hexadecimal).

```
MODIFY JOBNAME,DEBUG=(JES,IPADDR(12AB:0:0:CD30::/60))
```

Usage

- The specification of the trace on the MODIFY command is *not* additive. That is, the trace setting will be that of the last MODIFY command as shown in the following examples.
  - Using DEBug:

    ```
    MODIFY FTPDJG1,DEBUG=(NONE)
    +EZYFT82I Active traces: NONE
    MODIFY FTPDJG1,DEBUG=(CMD)
    +EZYFT82I Active traces: CMD
    MODIFY FTPDJG1,DEBUG=(FSC,USERID(USER33))
    +EZYFT82I Active traces: FSC(1)
    +EZYFT89I Userid filter: USER33
    MODIFY FTPDJG1,DEBUG=(SOC)
    +EZYFT82I Active traces: SOC(1)
    ```
  - Using DUMP:

    ```
    MODIFY FTPDJG1,DUMP=(NONE)
    +EZYFT83I Active dumpIDs: NONE
    MODIFY FTPDJG1,DUMP=(21)
    +EZYFT83I Active dumpIDs: 21
    MODIFY FTPDJG1,DUMP=(22)
    +EZYFT83I Active dumpIDs: 22
    ```
- The DUMP keyword can be used as shown in the following example:

  ```
  modify jobname,DUMP=(SQL,SOC)      ;sets all SQL and SOC DUMP ID's

  modify jobname,DUMP=(NONE)         ;resets all DUMP ID's

  modify jobname,DUMP=(Xnn)          ;resets  DUMP ID nn where nn is
                                     ;a number between 1 and 99

  modify jobname,DUMP=(XFSC)         ;resets all DUMP ID's 20 to 49

  modify jobname,DUMP=(XSOC)         ;resets all DUMP ID's 50 to 59

  modify jobname,DUMP=(XJES)         ;resets all DUMP ID's 60 to 69

  modify jobname,DUMP=(XSQL)         ;resets all DUMP ID's 70 to 79

  modify jobname,DUMP=(NONE,JES,X61) ;resets all ID's and
                                     ;then sets all JES DUMP ID's
                                     ;except number 61
  ```
- The `modify jobname,UTRACE` command that was supported in releases prior to z/OS V1R2 is not supported in this release. However, its function can be replaced with the following pair of commands:

  ```
  MODIFY jobname,DEBUG=(ALL,USERID(USER33))
  MODIFY jobname,DUMP=(ALL,USERID(USER33))
  ```

  Do not specify the ALL parameter on a routine basis, because it can produce an extensive amount of trace data.

- The `modify jobname,NOUTRACE` command that was supported in releases prior to z/OS V1R2 is not supported in this release. If complete tracing was activated as suggested above, then the tracing can be stopped using the following pair of commands:

```
MODIFY jobname,DEBUG=(NONE)
MODIFY jobname,DUMP=(NONE)
```

Context

For additional information see z/OS Communications Server: IP Diagnosis Guide.

MODIFY command: IKE server

You can use the operator console and the MODIFY command to control IKE server functions.

Format

```
├──┬─MODIFY─┬──procname,DISPLAY──────────────────────────────────────────┤
   └─F──────┘
```

```
├──┬─MODIFY─┬──procname,REFRESH──┬──────────────────────┬────────────────┤
   └─F──────┘                    ├─,FILE='filename'─────┤
                                 └─,FILE=//'filename'───┘
```

Parameters

procname
The member name of the cataloged procedure that is used to start the IKE server (IKED).

DISPLAY
Displays the configuration values that are currently being used by the IKE server.

REFRESH
Indicates that the IKE server configuration file should be reread. Not all IkeConfig parameters can be updated using this command. See the individual IkeConfig statement parameter descriptions for information about which parameters can be dynamically changed. See IkeConfig statement information in the z/OS Communications Server: IP Configuration Reference for details.

FILE
Indicates the name and location of the IKE server configuration file that is to be read. The *filename* value must be a fully qualified z/OS UNIX file name or an MVS data set name. You must enclose a z/OS UNIX file name in single quotation marks ('). MVS data set names must begin with two forward slashes (//) and you must enclose the data set name in single quotation marks ('). If the FILE parameter is omitted, the normal search order for locating the configuration data set or file applies. See the steps for configuring the IKE daemon in the z/OS Communications Server: IP Configuration Guide for information about the search order. This option is valid only when it is specified with the REFRESH parameter. If you omit this option, the IKE server rereads the configuration file with which it was started.

Examples

The following example displays the configuration values that are currently being used by the IKE server.

```
f iked,display

EZD1158I DISPLAY IKE CONFIGURATION
DISPLAY IKE configuration parameters:
  Values loaded from /etc/security/iked.conf
  IkeSyslogLevel = 1
  PagentSyslogLevel = 0
  SMF119 = NONE
  Keyring = iked/keyring
  IkeRetries = 6
  IkeInitWait = 2
  FIPS140 = no
  Echo = no
  PagentWait = 0
  NssWaitLimit = 5
  NssWaitRetries = 3
  IKE configuration contains no SupportedCertAuth labels.
  NetworkSecurityServer   = 10.81.5.5
                 Port     = 4159
                 Identity = 10.81.5.5
  IKE configuration contains no NetworkSecurityServerBackup info.
  NssStackConfig 1: Stack = TCPCS2
                    Client name = CLIENT2
                    Userid name = USER1
                    AuthBy = Password
                    Cert Service: Enabled
                    RemoteMgmtService: Enabled
```

MODIFY command: NCPROUTE

You can control most of the functions of the NCPROUTE address space from the operator console using the MODIFY command. The following list shows the correct syntax and valid parameters for the NCPROUTE address space.

Use the MODIFY command to pass parameters to the NCPROUTE address space.

Format

```
►►──┬─MODIFY─┬──procname──,──┬─────────┬──┬─PARMS=──parms──────┬──,──C=──client──────►◄
    └─F──────┘               └─QUERY───┘  ├─PROFILE───────────┤
                                          ├─GATEWAYS──────────┤
                                          ├─GATEWAYS,DELETE───┤
                                          └─TABLES────────────┘
```

Parameters

procname
: The member name of the cataloged procedure used to start the NCPROUTE server.

QUERY
: Queries the current target client NCP name or IP address.

parms
: Any one or more of the following separated by a space. Enclosing the *parms* specified in single quotation marks or preceding by a slash (/) is optional.

-g Enable default router.

-gq Disable default router.

-f Flush all indirect routes known from IP routing tables.

-fh Flush all indirect host routes known from IP routing tables.

-h Include host routes in addition to network-specific router for the RIP responses.

-hq Disable supply host routes.

-s Enable supply routing information.

-sd Enable supply default route only.

-sdq Disable supply default route only.

-sl Enable supplying of only local (directly connected) routes.

-slq Disable supplying of only local (directly connected) routes.

-sq or -q
Disable supply routing information.

-t Enable or disable traces. Up to 4 -t *parms* are allowed.

-tq Disable all traces.

-dp Enable debug packets trace.

-dq Disable all debug traces.

`PROFILE`
Reread the NCPROUTE PROFILE data set.

`GATEWAYS`
Reread the NCP client GATEWAYS data set member. If *,DELETE* is specified, all routes listed in the data set are deleted.

`TABLES`
Displays NCPROUTE internal IP routing and interface tables for diagnosis.

`client`
The target client NCP name or IP address. A value of 0 indicates all clients. The default will be the first client that has an established session with NCPROUTE. This parameter can be issued at once to indicate that NCPROUTE is to process modify commands for this client or for all clients. If C=0 is specified or if NCPROUTE does not have any active sessions with its clients, then only the parameters PARMS= and PROFILE are allowed to be processed.

Examples

```
F NCPROUT,GATEWAYS,c=NCP4
F NCPROUT,PARMS=-t -t -t -t,c=NCP1
F NCPROUT,PARMS=-tq, c=9.67.116.65
F NCPROUT,PARMS,c=10.1.1.99
F NCPROUT,PROFILE
F NCPROUT,PARMS=-tq
F NCPROUT,GATEWAYS,DELETE
F NCPROUT,PARMS,c=0
F NCPROUT,PARMS='/ -s -g'
F NCPROUT,PARMS=-h,PROFILE,GATEWAYS
```

Usage

Consider the following when coding the parms:

- Enclosing quotation marks for the parms are optional.
- Enclosing / for the parms is optional for example, parms=/-t -t).
- If the c= parameter cannot be specified in one command, issue the modify command with this parameter alone, following another modify command for other parameters.
- For -f or -fh parameters, only the indirect routes known by NCPROUTE are flushed:

Table 9 shows how the above parameters affect the advertising algorithm for routes in RIP responses to adjacent routers.

Note: The modify parameters correspond to the parameters in the OPTIONS statement of NCPROUTE Gateways data set.

Table 9. NCPROUTE Modify parameters

Parameter	NCPROUTE GATEWAY option	Host routes	Network routes	Advertise as default router	Local routes	Unreachable routes
-g	default router yes	No	Yes	Yes	Yes	Yes
-h	Supply local hosts	Yes	Yes	No	Yes	Yes
-s	Supply on	No	Yes	No	Yes	Yes
-sd	Supply default route	No	No	Yes	No	Yes
-sl	supply locals	No	No	No	Yes	Yes
-sq or -q	supply off	No	No	No	No	No
None	None	No	Yes	No	Yes	Yes

MODIFY command: Network security services server

You can use the operator console and the MODIFY command to control the network security services (NSS) server functions.

Format

```
├──┬─MODIFY─┬──procname,DISPLAY──┬──────────────────┬──────────────────────────────┤
   └─F──────┘                     └─'-,URLCACHE-'─┘
```

```
├──┬─MODIFY─┬──procname,REFRESH──┬─────────────────────────┬──────────────────────┤
   └─F──────┘                     ├─,FILE='filename'───────┤
                                  └─,FILE=//'filename'─────┘
```

Parameters

procname
The member name of the cataloged procedure that is used to start the network security services daemon (NSSD).

DISPLAY
Displays configuration values that are currently being used by the NSS server.

URLCACHE
Displays the current contents of the URL cache instead of displaying the configuration information. For each URL that has data cached, this command displays the type of data (Cert, Bundle, or CRL), the expiration date and time of the cache entry, and the URL for which data is cached.

REFRESH
Indicates that the NSS server configuration file should be reread and any cached certificate URL data should be flushed. See the Network security services server information in z/OS Communications Server: IP Configuration Reference for more information.

FILE
Indicates the name and location of the network security services (NSS) server configuration file that is to be read. The *filename* value must be a fully qualified z/OS UNIX file name or an MVS data set name. You must enclose a z/OS UNIX file name in single quotation marks ('). MVS data set names must begin with two forward slashes (//) and you must enclose the data set name in single quotation marks ('). If the FILE parameter is omitted, the normal search order for locating the configuration data set or file applies. See the steps for configuring the NSS server in the z/OS Communications Server: IP Configuration Guide for information about the search order. This option is valid only when it is specified with the REFRESH parameter. If you omit this option, the NSS server rereads the configuration file with which it was started.

Examples

The following example displays the configuration values that are currently being used by the NSS server.

```
f nssd,display

EZD1386I DISPLAY NSS CONFIGURATION
DISPLAY Network Security Server Configuration Parameters:
    Port        = 4159
    SyslogLevel = 255     (0x00ff)
    KeyRing     = "nssd/keyring"
    ----------------------------------
    Discipline IPSec        = Enabled
    Discipline XMLAppliance = Enabled
    ----------------------------------
  IPSec Discipline Configuration Parameters:
    FIPS140     = No
    URLCacheInterval = 10080
    There are 2 CertificateURL and CertificateBundleURL entries:
      Type   Label                            URL
      ------ -------------------------------- -------------------------
      Cert   Cert1                            http://example.com/cert1.der
      Bundle Root1 Chain MVSA Cert5           http://example.com/certbndl2.bndl
```

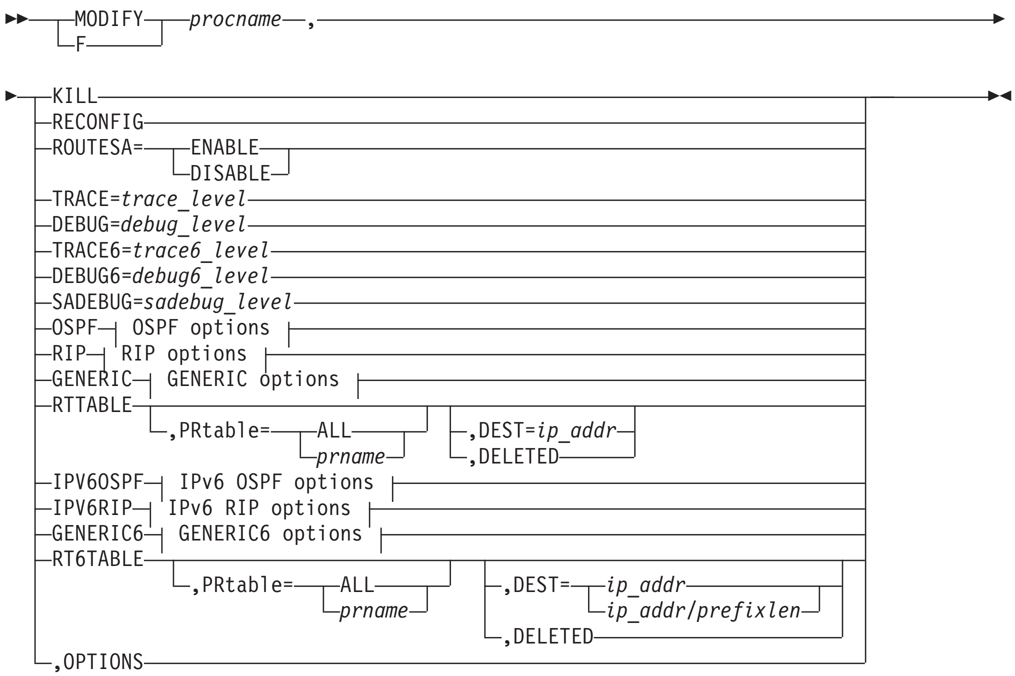
The following example displays the current URL cache information.

```
f nssd,display,urlcache

EZD1389I DISPLAY NSS URLCACHE:
URL Cache:
Type   Expiration          URL
------ ------------------- -------------------------
CRL    2010/02/04 12:48:12 HTTP://EXAMPLE.COM:80/crl.der
Cert   2010/02/04 12:50:36 HTTP://EXAMPLE.COM:80/cert2.der
2 URL Cache entries displayed.
```

MODIFY command: OMPROUTE

You can control OMPROUTE from the operator console using the MODIFY command.

Format



OSPF options:

```
├──┬─,LIST──┬─,ALL─────────┬──────────────────────────┬───────────────────────┤
   │        ├─,AREAS───────┤                          │
   │        ├─,InterFaceS──┤                          │
   │        ├─,NBMA────────┤                          │
   │        ├─,NeighBoRS───┤                          │
   │        └─,VLINKS──────┘                          │
   ├─┤ LSA command ├──────────────────────────────────┤
   ├─,AREASUM─────────────────────────────────────────┤
   ├─,EXTERNAL────────────────────────────────────────┤
   ├─,DATABASE──┬─────────────────────┬───────────────┤
   │            └─,AREAID=area_id─────┘               │
   ├─,DBSIZE──────────────────────────────────────────┤
   ├─,InterFace──┬──────────────────────────────────┬─┤
   │             └─,NAME=if_name──┬───────────────┬─┘ │
   │                              ├─,ACTIVATE─────┤   │
   │                              └─,SUSPEND──────┘   │
   ├─,NeighBoR──┬─────────────────────┬───────────────┤
   │            └─,IPADDR=ip_addr─────┘               │
   ├─,ROUTERS─────────────────────────────────────────┤
   ├─,STATiStics──────────────────────────────────────┤
   └─,WEIGHT──,NAME=name──,COST=cost──────────────────┘
```

LSA command:

```
├──,LSA──,LSTYPE=ls_type──,LSID=lsid──────────────────────────────►

►─,ORIGinator=ad_router──┬─────────────────────┬──────────────────┤
                         └─,AREAID=area_id─────┘
```

RIP options:

```
├──┬─,LIST──┬─,ALL─────────┬────────────┬─────────────────────────┤
   │        ├─,InterFaceS──┤            │
   │        └─,ACCEPTED────┘            │
   ├─,InterFace──┬─────────────────┬────┤
   │             └─,NAME=if_name───┘    │
   └─FILTERS────────────────────────────┘
```

GENERIC options:

```
├──┬─,LIST──┬─,ALL─────────┬──┬───────────────────────────────────┤
   │        └─,InterFaceS──┘  │
   └─,InterFace───────────────┘
```

IPv6 OSPF options:

```
├──┬─,ALL──────────────────────────────────────────────────┬──────────────────────────┤
   ├─,AREASUM──────────────────────────────────────────────┤
   ├─,InterFace──┬─────────────────────────────────────┬───┤
   │             ├─,NAME=──if_name──┬───────────────┬──┤   │
   │             │                  ├─,ACTIVATE─┤   │  │   │
   │             │                  └─,SUSPEND──┘   │  │   │
   │             └─,ID=──if_id──┬───────────────┬──────┘   │
   │                            ├─,ACTIVATE─┤              │
   │                            └─,SUSPEND──┘              │
   ├─,VLINK──┬─────────────────────────┬───────────────────┤
   │         └─,ENDPT=router-id─┘                          │
   ├─,NeighBoR──┬──────────────────────────────────────┬───┤
   │            └─,ID=router-id──┬──────────────────┬──┘   │
   │                             └─,IFNAME=if_name──┘      │
   ├─,DBSIZE───────────────────────────────────────────────┤
   ├─┤ IPv6 LSA command ├──────────────────────────────────┤
   ├─,EXTERNAL─────────────────────────────────────────────┤
   ├─,DATABASE──┬─────────────────────┬────────────────────┤
   │            └─,AREAID=area_id─┘                        │
   ├─,ROUTERS──────────────────────────────────────────────┤
   ├─,STATiStics───────────────────────────────────────────┤
   └─,WEIGHT──,NAME=name──,COST=cost───────────────────────┘
```

IPv6 LSA command:

```
├─,LSA──,LSTYPE=ls_type──,LSID=lsid──,ORIGinator=ad_router──┬──────────────────┬──┬──────────────────┬──────┤
                                                            └─,AREAID=area_id─┘  └─,IFNAME=if_name─┘
```

IPv6 RIP options:

```
├──┬─,ALL────────────────────────────┬───────────────────────────────────┤
   ├─,ACCEPTED───────────────────────┤
   ├─,InterFace──┬─────────────────┬─┤
   │             └─,NAME=if_name─┘   │
   └─,FILTERS────────────────────────┘
```

GENERIC6 options:

```
├──┬─,ALL────────────────────────────┬───────────────────────────────────┤
   └─,InterFace──┬─────────────────┬─┘
                 └─,NAME=if_name─┘
```

Parameters

procname
: The name of the member in a procedure library that was used to start OMPROUTE.

OPTIONS
: Specifies that the global configuration options information is to be displayed.

KILL
: Stop the OMPROUTE function.

RECONFIG
: Reread the OMPROUTE configuration file. This command ignores all

statements in the configuration file except new OSPF_Interface, RIP_Interface, Interface, IPv6_RIP_Interface, and IPv6_Interface statements.

Rule: If you do not have GLOBAL_OPTIONS IGNORE_UNDEFINED_INTERFACES=YES coded in your OMPROUTE configuration, these new configuration statements must be reread from the configuration file by using this command before the interface is first configured to the TCP/IP stack. If you have coded GLOBAL_OPTIONS IGNORE_UNDEFINED_INTERFACES=YES in your OMPROUTE configuration file, you can use OMPROUTE reconfiguration to add a definition for an interface that has been defined to the stack but that is ignored by OMPROUTE. However, OMPROUTE does not associate the interface with the new definition until the interface has been deleted from the stack and then re-added.

ROUTESA=ENABLE|DISABLE
Enable or disable the OMPROUTE subagent.

Note: To change any other value on the ROUTESA_CONFIG statement, the OMPROUTE application must be recycled.

TRACE=*trace_level*

Start, stop, or change the level of OMPROUTE tracing for initialization and IPv4 routing protocols. The different trace levels available and their descriptions are as follows:

TRACE=0
Turns off OMPROUTE tracing.

TRACE=1
Gives all the informational messages.

TRACE=2
Gives the informational messages plus formatted packet tracing.

Attention: OMPROUTE tracing affects OMPROUTE performance and might require increasing the Dead_Router_Interval on OSPF interfaces to keep neighbor adjacencies from collapsing.

DEBUG=*debug_level*
Level of debugging for OMPROUTE to use for initialization and IPv4 routing protocols. The following values are valid:

- DEBUG=0 turns off OMPROUTE IPv4 and initialization debugging.
- DEBUG=1 provides internal debug messages.
- DEBUG=2 provides the same information as DEBUG=1 plus hexadecimal packet tracing.
- DEBUG=3 provides the same information as DEBUG=2 plus module entry and exit.
- DEBUG=4 provides the same information as DEBUG=3 plus task add and run.

TRACE6=*trace6_level*
Start, stop, or change the level of OMPROUTE tracing for IPv6 routing protocols. The different trace levels available and their descriptions are as follows:

TRACE6=0
Turns off OMPROUTE tracing.

TRACE6=1
Gives all the informational messages.

TRACE6=2
Gives the informational messages plus formatted packet tracing.

Attention: OMPROUTE tracing affects OMPROUTE performance and might require increasing the Dead_Router_Interval on OSPF interfaces to keep neighbor adjacencies from collapsing.

DEBUG6=*debug6_level*
Level of debugging for OMPROUTE to use for IPv6 routing protocols. The following values are valid :

- DEBUG=0 turns off OMPROUTE IPv4 and initialization debugging.
- DEBUG=1 provides internal debug messages.
- DEBUG=2 provides the same information as DEBUG=1 plus hexadecimal packet tracing.
- DEBUG=3 provides the same information as DEBUG=2 plus module entry and exit.
- DEBUG=4 provides the same information as DEBUG=3 plus task add and run.

SADEBUG=*sadebug_level*
Level of debugging for OMPROUTE subagent to use.

OSPF
Specifies that OSPF information is to be displayed.

LIST
Specifies that OSPF information is to be displayed as defined in the OMPROUTE configuration file.

ALL
Displays a comprehensive list of all configuration information.

AREAS
Displays all information concerning configured OSPF areas and their associated ranges.

InterFaceS
Displays, for each OSPF interface, the IP address and configured parameters as coded in the OMPROUTE configuration file.

NBMA
Displays the interface address and polling interval related to interfaces connected to non-broadcast multi-access networks.

NeighBoRS
Displays the configured neighbors on non-broadcast networks.

VLINKS
Displays all virtual links that have been configured with this router as the endpoint.

LSA
Displays the contents of a single link state advertisement contained in the OSPF database.

A link state advertisement is defined by its

- Link state type (**LSTYPE=***ls_type*)
- Link state ID (**LSID=***lsid*)

- Advertising router (**ORIGinator=***ad_router*).

There is a separate link state database for each OSPF area. **AREAID=***area_id* on the command line tells the software which database you want to search. The different kinds of advertisements, which depend on the value given for link-state-type, are:

Router links (LSTYPE=1)
Describe the collected states of a router interface attached to a router.

Network links (LSTYPE=2)
Describe the set of routers attached to a network.

Summary link, IP network (LSTYPE=3)
Describe interarea routes to networks.

Summary link, ASBR (LSTYPE=4)
Describe interarea routes to AS boundary routers.

AS external link (LSTYPE=5)
Describe routes to destinations external to the Autonomous System.

Note: The `ORIGINATOR` needs to be specified only for link-state-types three, four, and five. The AREAID value needs to be specified for all link-state-types except five.

Link State IDs, originators (specified by their router IDs), and area IDs take the same format as IP addresses. For example, the backbone area can be entered as `0.0.0.0`

AREASUM
Displays the statistics and parameters for all OSPF areas attached to the router.

EXTERNAL
Displays the AS external advertisements belonging to the OSPF routing domain. One line is printed for each advertisement.

DATABASE,AREAID=*area_id*
Displays a description of the contents of a particular OSPF area link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. If AREAID is not specified, the database from area 0.0.0.0 will be displayed.

DBSIZE
Displays the number of LSAs currently in the link state database, categorized by type

InterFace,NAME=*if_name*
Displays current, run-time statistics and parameters related to OSPF interfaces. If a `NAME=`*if_name* parameter is omitted, a single line is printed summarizing each interface. If a `NAME=`*if_name* parameter is specified, detailed statistics for that interface will be displayed.

ACTIVATE
Activate an OSPF interface that is in suspend state to allow adjacency formations with neighbors over this interface. This parameter is not applicable for static and dynamic VIPA interfaces. If this is a LAN interface and there is another alternate redundant interface on this same LAN segment that is the primary OSPF interface, this interface

becomes a backup interface. This command does not force the activated interface to take over or resume the primary OSPF interface role for the LAN segment.

SUSPEND

Suspend an active OSPF interface that is not in DOWN or SUSPEND state so that adjacency formations with neighbors over this interface are stopped or not allowed. This forces adjacency attempts with neighbors over an alternate redundant interface if this is a LAN interface and an alternate interface is available. Existing connections that use static routes over the suspended interface are not disrupted. This parameter is not applicable for static and dynamic VIPA interfaces. If a TCP/IP stack is recycled while an interface is in suspended state, the interface state is reset after the recycle.

NeighBoR,IPADDR=*ip_addr*

Displays the statistics and parameters related to OSPF neighbors. If an IPADDR=*ip_addr* parameter is omitted, a single line is printed summarizing each neighbor. If an IPADDR=*ip_addr* parameter is given, detailed statistics for that neighbor are displayed.

ROUTERS

Displays all routes to other routers that have been calculated by OSPF and are currently present in the routing table.

STATiStics

Displays statistics generated by the OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization. Many of the fields displayed are confirmation of the OSPF configuration.

WEIGHT

Dynamically change the cost of an OSPF interface. This new cost is flooded quickly throughout the OSPF routing domain, and modifies the routing immediately.

The cost of the interface reverts to its configured value whenever OMPROUTE is restarted. To make the cost change permanent, you must reconfigure the appropriate OSPF interface in the configuration file. This command can be issued only for an OSPF interface that is active in the TCP/IP stack.

NAME=*name*

Name of the OSPF interface the new cost affects.

COST=*cost*

New cost value for the OSPF interface.

RIP

Specifies that RIP information is to be displayed.

LIST

Specifies that RIP information is to be displayed as defined in the OMPROUTE configuration file.

ALL

Display all RIP-related configuration information.

InterFaceS

Display IP addresses and configured parameters for each RIP interface.

ACCEPTED
Displays the routes to be unconditionally accepted, as configured with the ACCEPT_RIP_ROUTE statement.

InterFace,NAME=*if_name*
Displays statistics and parameters related to RIP interfaces. If a NAME=*if_name* parameter is omitted, a single line is printed summarizing each interface. If a NAME=*if_name* parameter is given, detailed statistics for the specified interface (*if_name*) are displayed.

FILTERS
Displays the Global RIP filters.

GENERIC
Specifies that IPv4 information not related to a specific routing protocol is to be displayed.

LIST
Specifies that information is to be displayed as defined in the OMPROUTE configuration file.

ALL
Displays all IPv4 information that is not related to a specific routing protocol.

InterFaceS
Lists all generic IPv4 interfaces that are defined to OMPROUTE using INTERFACE statements.

InterFace
Displays statistics and parameters related to IPv4 generic interfaces that are known to TCP/IP.

RTTABLE

Displays routes in an OMPROUTE IPv4 routing table. If this option is used without the PRtable option, the routes that are displayed are from the main routing table.

DEST=*ip_addr*
Displays the routes to a particular destination. When multiple equal-cost routes exist, use this option to obtain a list of the next hops. You cannot use this option with the DELETED option.

PRtable=ALL
Displays routes in all of the OMPROUTE IPv4 policy-based routing tables. The dynamic routing parameters configured to the Policy Agent for a table are displayed following the routes for the table.

PRtable=*prname*
Displays routes in the specified OMPROUTE IPv4 policy-based routing table. The dynamic routing parameters configured to the Policy Agent for the table are displayed following the routes for the table.

DELETED
Displays information about routes that have been deleted from the OMPROUTE routing table and that have not been replaced. You cannot use this option with the DEST=*ip_addr* option.

Results:

- If the RIP protocol is running, deleted routes are displayable for only 3 minutes after deletion. After 3 minutes have elapsed they are garbage collected by RIP and are no longer displayable.
- If a policy-based routing table is configured to the Policy Agent with no IPv4 dynamic routing parameters, OMPROUTE has no knowledge of that routing table for IPv4. The routing table does not appear in the display of OMPROUTE IPv4 routing tables. The routing table does not appear in the display of OMPROUTE routing tables.
- Only active policy-based routing tables appear in the display of OMPROUTE routing tables. A policy-based routing table is active if it is referenced by an active routing rule and its associated action.
- This option displays the contents of the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see "DISPLAY TCPIP,,NETSTAT" on page 9.

IPV6OSPF

Specifies that IPv6 OSPF information is to be displayed.

ALL

Displays a comprehensive list of IPv6 OSPF information.

AREASUM

Displays the statistics and parameters for all IPv6 OSPF areas attached to the router.

InterFace,NAME=*if_name* **or InterFace,ID=***if_id*

Displays current, run-time statistics and parameters related to IPv6 OSPF interfaces. If the NAME= and ID= parameters are omitted, a single line is printed summarizing each interface. If the NAME= or ID= parameter is specified, detailed statistics for that interface will be displayed.

VLINK,ENDPT=*router-id*

Displays current, run-time statistics and parameters related to IPv6 OSPF virtual links. If the ENDPT= parameter is omitted, a single line is printed summarizing each virtual link. If the ENDPT= parameter is specified, detailed statistics for that virtual link will be displayed.

NeighBoR,ID=*router-id***,IFNAME=***if_name*

Displays the statistics and parameters related to IPv6 OSPF neighbors.

- If the ID= parameter is omitted, a single line is printed summarizing each neighbor.
- If the ID= parameter is given, detailed statistics for that neighbor are displayed.
- If the neighbor specified by the ID= parameter has more than one neighbor relationship with OMPROUTE (for example if there are multiple IPv6 OSPF links connecting them), the IFNAME= parameter can be used to specify which link's adjacency to examine (for an adjacency over a virtual link, specify IFNAME=*).

DBSIZE

Displays the number of LSAs currently in the IPv6 OSPF link state database, categorized by type.

LSA

Displays the contents of a single link state advertisement contained in the IPv6 OSPF database. A link state advertisement is defined by its:

- Link state type (LSTYPE=*ls_type*, where *ls_type* is one of the hexadecimal link state type values listed below).
- Link state ID (LSID=*lsid*).
- Advertising router (ORIGinator=*ad_router*).

Each interface has its own set of link LSAs (LSTYPE=0008). IFNAME=interface_name on the command line indicates which link's LSA you want to display.There is also a separate link state database for each IPv6 OSPF area. AREAID=area_id on the command line indicates which database you want to search. If you do not specify which area to search, the backbone (0.0.0.0) area will be searched. The different kinds of advertisements, which depend on the value given for link state type, are:

Router LSA (LSTYPE=2001)

The complete collection describes the state and cost of the router's interfaces to the area. Each router in an area originates one or more Router LSAs.

Network LSA (LSTYPE=2002)

Originated by the Designated Router of each multiaccess link (i.e., LAN) in the area which supports two or more routers. Describes the set of routers attached to the link, including the Designated Router.

Inter-Area Prefix LSA (LSTYPE=2003)

Originated by an area border router. Describes the route to an IPv6 address prefix that belongs to another area.

Inter-Area Router LSA (LSTYPE=2004)

Originated by an area border router. Describes the route to an AS boundary router that belongs to another area.

AS External LSA (LSTYPE=4005)

Originated by an AS boundary router. Describes the route to a destination external to the IPv6 OSPF Autonomous System.

Link LSA (LSTYPE=0008)

Originated by routers for each link to which they are attached. Provides the router's link-local address, provides a list of IPv6 address prefixes for the link, and asserts a set of options for the Network LSA that will be originated for the link.

Intra-Area Prefix LSA (LSTYPE=2009)

Originated by routers to advertise one or more IPv6 address prefixes that are associated with the router itself, an attached stub network segment, or an attached transit network segment.

Requirements:

1. Specify the AREAID value for all link state types except AS External LSA.

   Note: The AREAID value defaults to the backbone (0.0.0.0) area if not specified.
2. Specify the IFNAME value for Link LSAs (LSTYPE=0008).

3. Originators (specified by their router IDs) and area IDs are specified in dotted-decimal format. For example, the backbone area is entered as 0.0.0.0.

EXTERNAL
Displays the AS external LSAs belonging to the IPv6 OSPF routing domain. One line is printed for each advertisement.

DATABASE,AREAID=*area_id*
Displays the contents of a particular IPv6 OSPF area link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. If AREAID is not specified, the database from area 0.0.0.0 will be displayed.

ROUTERS
Displays all routes to other routers that have been calculated by IPv6 OSPF and are currently present in the routing table.

STATISTICS
Displays statistics generated by the IPv6 OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization.

WEIGHT
Dynamically change the cost of an IPv6 OSPF interface. This new cost is flooded quickly throughout the IPv6 OSPF routing domain, and modifies the routing immediately. The cost of the interface reverts to its configured value whenever OMPROUTE is restarted. To make the cost change permanent, you must reconfigure the appropriate IPv6 OSPF interface in the OMPROUTE configuration file.

NAME=*name*
Name of the IPv6 OSPF interface the new cost affects.

COST=*cost*
New cost value for the IPv6 OSPF interface.

IPV6RIP
Specifies the IPv6 RIP information.

ALL
Displays all IPv6 RIP-related information.

ACCEPTED
Displays the routes to be unconditionally accepted, as configured with the IPV6_ACCEPT_RIP_ROUTE statement.

InterFace,NAME=*if_name*
Displays statistics and parameters related to IPv6 RIP interfaces. If the NAME=*if_name* parameter is omitted, a single line is printed summarizing each interface. If the NAME=*if_name* parameter is given, detailed statistics for the specified interface (*if_name*) are displayed.

FILTERS
Displays the Global IPv6 RIP filters.

GENERIC6
Specifies IPv6 information not related to a specific dynamic routing protocol.

ALL
Displays all IPv6 information not related to a specific routing protocol.

InterFace,NAME=*if_name*

Displays statistics and parameters related to IPv6 generic interfaces that are known to TCP/IP or defined to OMPROUTE with IPV6_INTERFACE statements. If the NAME=*if_name* parameter is omitted, a single line is printed summarizing each interface. If the NAME=*if_name* parameter is given, detailed statistics for the specified interface (*if_name*) is displayed.

RT6TABLE

Displays routes in an OMPROUTE IPv6 routing table. If this option is used without the PRtable option, the routes that are displayed are from the main routing table.

DEST=*ip_addr/prefixlen*

Displays information about a particular route. When multiple equal-cost routes exist, use this option to obtain a list of the next hops. You cannot use this option with the DELETED option.

PRtable=ALL

Displays routes in all of the OMPROUTE IPv6 policy-based routing tables. The dynamic routing parameters that are configured to the policy agent for a table are displayed following the routes for the table.

PRtable=*prname*

Displays routes in the specified OMPROUTE IPv6 policy-based routing table. The dynamic routing parameters that are configured to the policy agent for the table are displayed following the routes for the table.

DELETED

Displays information about IPv6 routes that have been deleted from the OMPROUTE routing table and that have not been replaced. You cannot use this option with the DEST=*ip_addr/prefixlen* option.

Results:

- If the RIP protocol is running, deleted routes are displayable for only 3 minutes after they are deleted. After 3 minutes have elapsed they are garbage collected by RIP and are no longer displayable.
- If a policy-based routing table is configured to the policy agent with no IPv6 dynamic routing parameters, OMPROUTE has no knowledge of that routing table for IPv6. The routing table is not included in the display of OMPROUTE IPv6 routing tables.
- Only active policy-based routing tables are included in the display of OMPROUTE routing tables. A policy-based routing table is active if an active routing rule and its associated action reference the routing table.
- This option displays the contents of the working tables that are used by OMPROUTE; it does not display the TCP/IP routing tables. The OMPROUTE routing tables might contain information that is different from the information in the TCP/IP routing tables. For more information about displaying the contents of the TCP/IP routing tables, see “DISPLAY TCPIP,,NETSTAT” on page 9.

Examples

You can use MODIFY OMPROUTE commands to perform the following functions:

- “Displaying OMPROUTE information” on page 204
- “Stopping OMPROUTE” on page 204
- “Rereading the configuration file” on page 204

- "Enabling or disabling the OMPROUTE subagent"
- "Changing the cost of OSPF links" on page 205

Displaying OMPROUTE information

You can use the MODIFY command to display information for OMPROUTE. For example, assume you have a *procname* of OMPROUT2 running on stack TCPCS2.

- To display the OMPROUTE IPv4 main routing table you can use:

  ```
  f omprout2,rttable
  ```
- To display ospf neighbors you can use:

  ```
  f omprout2,ospf,nbrs
  ```

See "DISPLAY TCPIP,,OMPROUTE" on page 23 for information about parameter descriptions and use.

Stopping OMPROUTE

OMPROUTE can be stopped in several ways:

- From MVS, issue P *procname* or MODIFY *procname*,KILL.

  If OMPROUTE was started from a cataloged procedure, *procname* is the member name of that procedure. If OMPROUTE was started from the z/OS shell, *procname* is *userid*X, where X is the sequence number set by the system. To determine the sequence number, from the SDSF LOG window on TSO, issue `/d omvs,u=`*`userid`*. This will show the programs running under the user ID *userid*. The *procname* value can also be set using the environment variable _BPX_JOBNAME and then starting OMPROUTE in the shell background.
- From a z/OS shell superuser ID, issue the kill command to the process ID (PID) associated with OMPROUTE. To find the PID, use one of the following methods:
  - From the MVS console, issue *D OMVS,U=userid*, or issue */D OMVS,U=userid* at the SDSF LOG window on TSO (where *userid* is the user ID that started omproute from the shell).
  - Issue the ps -ef command from the z/OS shell.
  - Write down the PID when you start OMPROUTE.

For information about the environment variable _BPX_JOBNAME, see z/OS UNIX System Services Planning. For information about the *D OMVS,U=userid* command, see z/OS MVS System Commands.

Rereading the configuration file

The MODIFY *procname*,RECONFIG command is used to reread the OMPROUTE configuration file. This command ignores all statements in the configuration file except new OSPF_Interface, RIP_Interface, Interface, IPv6_RIP_Interface, IPv6_Interface, IPv6_OSPF_Interface, and IPv6_OSPF (ROUTERID parameter only) statements.

Rule: These new configuration statements must be reread from the configuration file through this command prior to any new interfaces referred to by new OMPROUTE configuration statements being configured to the TCP/IP stack.

Enabling or disabling the OMPROUTE subagent

Use the MODIFY *procname*,ROUTESA=ENABLE command or the MODIFY *procname*,ROUTESA=DISABLE command to enable or disable the OMPROUTE subagent.

Note: To change any other value on the ROUTESA_CONFIG statement, the OMPROUTE application must be recycled.

The OMPROUTE subagent implements RFC 1850 (OSPF Version 2 Management Information Base) for the OSPF (Open Shortest Path First) Protocol. The ROUTESA_CONFIG statement is used in the OMPROUTE configuration file to configure the OMPROUTE subagent. For details about the ROUTESA_CONFIG statement, see ROUTESA_CONFIG statement information in the z/OS Communications Server: IP Configuration Reference.

Changing the cost of OSPF links

The cost of an OSPF interface can be dynamically changed using the `MODIFY procname,OSPF,WEIGHT,NAME=`*`name`*`,COST=`*`cost`* command for an IPv4 OSPF interface or the `MODIFY procname,IPV6OSPF,WEIGHT,NAME=`*`name`*`,COST=`*`cost`* command for an IPv6 OSPF interface. This new cost is flooded quickly throughout the OSPF routing domain, and modifies the routing immediately.

The cost of the interface reverts to its configured value whenever OMPROUTE is restarted. To make the cost change permanent, you must reconfigure the appropriate OSPF interface in the configuration file.

MODIFY command: Policy Agent

You can use the operator console and the MODIFY command to control the Policy Agent functions.

Format

```
►►──┬─MODIFY─┬──procname──,──────────────────────────────────────────────►
    └─F──────┘

►──┬─LOGLEVEL──,──LEVEL=n────────────────────────────┬──────────────────►◄
   ├─TRACE──,──LEVEL=t───────────────────────────────┤
   ├─DEBUG──,──LEVEL=d───────────────────────────────┤
   ├─MEMTRC──────────────────────────────────────────┤
   ├─QUERY───────────────────────────────────────────┤
   ├─REFRESH─────────────────────────────────────────┤
   ├─SRVLSTN─────────────────────────────────────────┤
   ├─UPDATE──────────────────────────────────────────┤
   └─MON──,──┬─DISPLAY─────────────────────────────┬─┘
             └─┬─START────┬──,──┬─ALL───────────┬──┘
               ├─RESTART──┤     ├─DMD───────────┤
               └─STOP─────┘     ├─IKED──────────┤
                                ├─NSSD──────────┤
                                ├─SYSLOGD───────┤
                                └─TRMD──┬─────────────┬┘
                                        └─,── P=image─┘
```

Parameters

procname
: The member name of the cataloged procedure used to start the Policy Agent.

LOGLEVEL,LEVEL=*n*
: Changes the Policy Agent LogLevel. The required log level is *n*. If *n* is not specified, then the current LogLevel remains the same. See LogLevel statement information in the z/OS Communications Server: IP Configuration Reference for details on how to define the Policy Agent LogLevel.

TRACE,LEVEL=*t*
: Changes the Policy Agent start option trace level. The required trace level is *t*.

If *t* is not specified, then the current trace level remains the same. See the Starting Policy Agent from the z/OS shell information in the z/OS Communications Server: IP Configuration Reference for details on valid Policy Agent trace levels.

Note: If Policy Agent was started with the trace option disabled, then the output destination of stderr will be closed. This option cannot later be enabled by using the MODIFY command.

DEBUG,LEVEL=*d*

Changes the Policy Agent start option debug level. The required debug level is *d*. If *d* is not specified, then the current debug level remains the same. See the Starting Policy Agent from the z/OS shell information in the z/OS Communications Server: IP Configuration Reference for details on valid Policy Agent debug levels.

MEMTRC

Causes the Policy Agent to dump the contents of the memory request buffer to the log file. This buffer is used when the -m startup option is specified, so if this option is not specified, the MEMTRC parameter has no effect.

QUERY

Displays the current LogLevel, debug level, and trace level in effect for the Policy Agent.

REFRESH

Triggers the Policy Agent to reread the configuration files, and, if requested, download objects from the LDAP server. Basically you download objects from the LDAP server only if a ReadFromDirectory statement is included in the configuration file. Note that policies are also refreshed if the SIGHUP signal is received by the Policy Agent. This signal can be sent using the UNIX `kill` command. If the FLUSH parameter was specified on the TcpImage or discipline configuration statement, the REFRESH command triggers FLUSH processing. One consequence of this is that policy statistics being collected in the TCPIP stack are reset, because FLUSH deletes and reinstalls all policies.

See FLUSH and PURGE considerations details in z/OS Communications Server: IP Configuration Guide for more information concerning the FLUSH/NOFLUSH and PURGE/NOPURGE parameters.

Tip: If you specify the Security Secure value on the ServicesConnection statement and the generated AT-TLS policy is installed successfully, then the MODIFY REFRESH command removes all AT-TLS policies, including the generated AT-TLS policy, if FLUSH is specified for AT-TLS. The AT-TLS policies, including the generated AT-TLS policy, are then reinstalled. The services connection might be unavailable until the generated AT-TLS policy is reinstalled.

SRVLSTN

Triggers the Policy Agent to restart the listen for services requestor connections and if required, to reinstall the generated AT-TLS policy. See ServicesConnection statement information in z/OS Communications Server: IP Configuration Reference for more details about configuring the ServicesConnection statement.

Tips:

- If you specify the Security Secure value on the ServicesConnection statement and the generated AT-TLS policy is installed successfully, use the MODIFY command with the SRVLSTN parameter to trigger the Policy Agent to reinstall the generated AT-TLS policy. Use this command when the contents of the key ring have changed, but the key ring name is unchanged.

- If you specify the Security Secure value on the ServicesConnection statement and the configured local or remote AT-TLS policies did not install successfully, use the MODIFY command with the SRVLSTN parameter to force the generated AT-TLS policy to be installed before the local or remote AT-TLS policies are installed. See the AT-TLS TCP/IP stack initialization access control information in z/OS Communications Server: IP Configuration Guide for more details about stack initialization access control.
- If the ImageName value that is configured on the ServicesConnection statement is not active when the ServicesConnection statement is processed, issue the MODIFY command with the SRVLSTN parameter after the TCP/IP image becomes active.

UPDATE

Triggers the Policy Agent to reread configuration files and, if requested, download objects from the LDAP server. Basically you download objects from the LDAP server only if a ReadFromDirectory statement is included in the configuration file. This command is different from the REFRESH command because Pagent only installs or removes from the stack as appropriate any new, changed, or deleted policies.

See FLUSH and PURGE considerations information in the in the z/OS Communications Server: IP Configuration Guide for more information concerning the FLUSH/NOFLUSH and PURGE/NOPURGE parameters.

MON

Send a command to an application that is being monitored by the Policy Agent.

DISPLAY

Display information about the set of applications, including whether or not they are being monitored, their status, and the associated TCP/IP stack name, if any.

START

Start a specified application or start all applications that are configured on the AutoMonitorApps statement to be started and stopped. Policy Agent starts the applications using the cataloged procedure and other parameters that are configured on the AutoMonitorApps statement.

Result: If the Policy Agent has stopped monitoring the applications because the applications failed to successfully start within the retry period that was specified on the AutoMonitorParms statement, Policy Agent resumes monitoring the running status of the applications.

ALL

Start all applications that are configured on the AutoMonitorApps statement.

DMD

Start the Defense Manager daemon (DMD).

IKED

Start the IKE daemon (IKED).

NSSD

Start the network security services daemon (NSSD).

SYSLOGD
Start the syslog daemon (syslogd).

TRMD
Start the traffic regulation management daemon (TRMD).

P=*image*
Specifies the name of the TCP/IP stack on which the TRMD application is running. If only one instance of TRMD is configured on the AutoMonitorApps statement, this parameter is optional.

RESTART
Stop and restart a specified application or stop and restart all applications that are configured on the AutoMonitorApps statement to be started and stopped. Policy Agent restarts the applications using the cataloged procedure and other parameters that are configured on the AutoMonitorApps statement.

ALL
Restart all applications that are configured on the AutoMonitorApps statement.

DMD
Restart the Defense Manager daemon (DMD).

IKED
Restart the IKE daemon (IKED).

NSSD
Restart the network security services daemon (NSSD).

SYSLOGD
Restart the syslog daemon (syslogd).

TRMD
Restart the traffic regulation management daemon (TRMD).

P=*image*
Specifies the name of the TCP/IP stack on which the TRMD application is running. If only one instance of TRMD is configured on the AutoMonitorApps statement, this parameter is optional.

STOP
Stop a specified application or stop all applications that are configured on the AutoMonitorApps statement to be started and stopped.

Result: Policy Agent stops monitoring the running status of the applications.

ALL
Stop all applications that are configured on the AutoMonitorApps statement.

DMD
Stop the Defense Manager daemon (DMD).

IKED
Stop the IKE daemon (IKED).

NSSD
Stop the network security services daemon (NSSD).

SYSLOGD
Stop the syslog daemon (SYSLOGD).

TRMD
Stop the traffic regulation management daemon (TRMD).

P=*image*
Specifies the name of the TCP/IP stack on which the TRMD application is running. If only one instance of TRMD is configured on the AutoMonitorApps statement, this parameter is optional.

Examples

The following example displays the status of applications that are monitored by the Policy Agent.

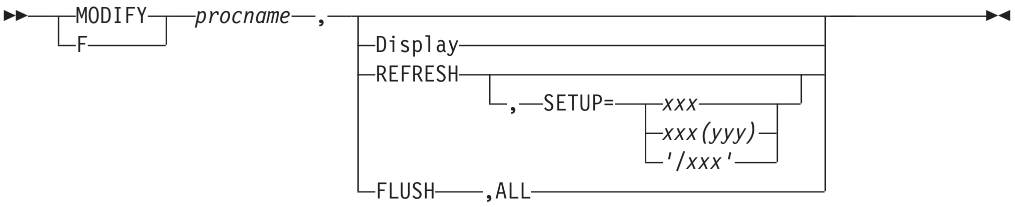
```
F PAGENT,MON,DISPLAY

EZD1587I PAGENT MONITOR INFORMATION
APPLICATION  MONITORED  JOBNAME  STATUS      TCP/IP STACK
DMD          NO         N/A      N/A         N/A
IKED         YES        IKED     ACTIVE      N/A
NSSD         YES        NSSD     RESTARTING  N/A
SYSLOGD      YES        SYSLOGD  ACTIVE      N/A
TRMD         YES        TRMD2    ACTIVE      TCPIP2
TRMD         YES        TRMD3    INACTIVE    TCPIP3
```

MODIFY command: Resolver address space

You can refresh the resolver address space from the operator console using the MODIFY command. Issue the REFRESH command to refresh the resolver address space and the DISPLAY command to display the current values of the resolver setup statements. Issue the FLUSH command to delete the contents of the resolver cache. You can also reset the current z/OS knowledge of name server capabilities by issuing the REFRESH command.

For a description of the resolver setup statements, see Resolver setup statements in z/OS Communications Server: IP Configuration Reference.

Format



Parameters

procname
The member name of the cataloged procedure used to start the resolver.

You can use the `Display OMVS,O` command to determine the *procname* value. The output displayed includes a line as follows:

```
RESOLVER PROC   = DEFAULT
```

If `DEFAULT` is displayed, then the *procname* value is `RESOLVER`. Any other value must be used as the *procname*.

Display
Displays the current resolver setup statement values. If the autonomic quiescing of unresponsive name server function is active, the resolver also displays the status of the name servers specified on NSINTERADDR statements in the global TCPIP.DATA file.

FLUSH
Deletes the contents of the resolver cache.

ALL
All resolver cache entries are deleted.

REFRESH
Causes applications to have their TCPIP.DATA information updated on their next resolver request after the refresh occurs, including local host tables (for example, etc/hosts, etc/ipnodes, HOSTS.SITEINFO, HOSTS.ADDRINFO, or ETC.IPNODES information). For more information about what TCPIP.DATA information can be updated, see Dynamically changing TCPIP.DATA statements in z/OS Communications Server: IP Configuration Reference.

Resets information about name server capabilities that was dynamically acquired. For example, a name server has been upgraded to support extension mechanisms for DNS (EDNS0). Issue the REFRESH command to reset the current information about the capability of the name server, forcing the z/OS resolver to dynamically determine the new capability.

Clears message EZD2037E from the network operator console. Issue the REFRESH command to restart the resolver unresponsive name server monitoring functions after the resolver had stopped them as a result of a system error.

Restriction: The REFRESH command resets information about all name servers.

The z/OS resolver uses EDNS0 to accept DNS UDP messages that have a length greater than 512 bytes when the name server also supports EDNS0. Using the less costly UDP protocol, EDNS0 results in improved DNS and resolver performance. See the information about Extension Mechanisms for DNS standards and the resolver in z/OS Communications Server: IP Configuration Guide for more details.

In some cases, the resolver statistics regarding name server responsiveness for all name servers are affected by a REFRESH command. This situation occurs when the REFRESH command is used to stop the name server monitoring functions or when the REFRESH command is used to switch between the autonomic quiescing of unresponsive name server function and the network operator notification function. In these situations, the resolver statistics are set to the value of 0 and the current unresponsive name server notification messages EZZ9308E or EZZ9311E are cleared from the operator console. If the REFRESH command does not result in any changes to the unresponsive name server monitoring function, the resolver statistics are not affected and the current unresponsive name server notification messages remain on the operator console. For information about when the statistics and messages are affected, see UNRESPONSIVETHRESHOLD statement in z/OS Communications Server: IP Configuration Reference.

If the REFRESH command is used to change the NSINTERADDR statement values coded in the TCPIP.DATA file specified on the GLOBALTCPIPDATA resolver setup statement, the resolver statistics regarding name server responsiveness for a single name server might be affected. If the REFRESH command is used to remove an unresponsive name server from the NSINTERADDR statements, the resolver will zero the resolver statistics for this name server, will stop sending resolver DNS polling queries to the name server, and will clear the EZZ9311E unresponsive name server notification message from the operator console. For information on the NSINTERADDR statement, see NSINTERADDR statement in z/OS Communications Server: IP Configuration Reference.

SETUP=

The contents of the specified resolver setup file are processed. Processing TCPIP.DATA statements and local host tables are updated at the next resolver request.

Result: If the resolver setup file contains unrecognized resolver setup statements, or setup statements with syntax errors, the MODIFY command fails. The resolver configuration is unchanged.

xxx

Identifies a specific MVS sequential data set. The data set must have an LRECL in the range 56–256. The record format can be either RECFM=F or RECFM=FB.

xxx(yyy)

Identifies a specific MVS PDS member. The PDS must have an LRECL in the range 56–256. The record format can be either RECFM=F or RECFM=FB.

'/xxx'

The full path name of the file must be specified and must begin with a slash (/) character. The single quotation marks (') are required around the complete z/OS UNIX file system name so that z/OS command processing passes the file name without changing it to uppercase.

Rule: If the single quotation mark notation is used to specify an MVS data set name, the data set name must be entered in uppercase.

Examples

The following example is the command and messages returned to display the current values. In this example, the AUTOQUIESCE operand on the UNRESPONSIVETHRESHOLD resolver setup statement was not specified:

```
f resolver,display

EZZ9298I DEFAULTTCPIPDATA - None
EZZ9298I GLOBALTCPIPDATA - SYS1.TCPPARMS(TCPDATA)
EZZ9298I DEFAULTIPNODES - USER55.ETC.IPNODES
EZZ9298I GLOBALIPNODES - None
EZZ9304I NOCOMMONSEARCH
EZZ9304I CACHE
EZZ9298I CACHESIZE - 200M
EZZ9298I MAXTTL - 2147483647
EZZ9298I UNRESPONSIVETHRESHOLD - 25
EZZ9293I DISPLAY COMMAND PROCESSED
```

The following example is the command and messages returned to display the current values. In this example, errors have been detected during initialization of the resolver address space.

```
f resolver,display

EZZ9298I DEFAULTTCPIPDATA - None
EZZ9298I GLOBALTCPIPDATA - SYS1.TCPPARMS(TCPDATA)
EZZ9298I DEFAULTIPNODES - USER55.ETC.IPNODES
EZZ9298I GLOBALIPNODES - None
EZZ9304I NOCOMMONSEARCH
EZZ9304I CACHE
EZZ9298I CACHESIZE - 200M
EZZ9298I MAXTTL - 2147483647
EZZ9298I UNRESPONSIVETHRESHOLD – 25
EZD2039I WARNINGS ISSUED DURING RESOLVER INITIALIZATION
EZZ9293I DISPLAY COMMAND PROCESSED
```

In the following example, the AUTOQUIESCE operand on the UNRESPONSIVETHRESHOLD resolver setup statement is coded:

```
f resolver,display

EZZ9298I DEFAULTTCPIPDATA - None
EZZ9298I GLOBALTCPIPDATA - SYS1.TCPPARMS(TCPDATA)
EZZ9298I DEFAULTIPNODES - USER55.ETC.IPNODES
EZZ9298I GLOBALIPNODES - None
EZZ9304I NOCOMMONSEARCH
EZZ9304I CACHE
EZZ9298I CACHESIZE - 200M
EZZ9298I MAXTTL - 2147483647
EZZ9298I UNRESPONSIVETHRESHOLD - 25
EZZ9304I AUTOQUIESCE
EZD2305I NAME SERVER 10.1.1.1
                      STATUS: ACTIVE          FAILURE RATE:   0%
EZD2305I NAME SERVER 10.2.2.2
                      STATUS: QUIESCED        FAILURE RATE:  60%
EZD2305I NAME SERVER 10.3.3.3
                      STATUS: ACTIVE          FAILURE RATE: *NA*
EZZ9293I DISPLAY COMMAND PROCESSED
```

The following example shows how to change some of current values in `user55.ressetup` setup file, such as changing `NOCOMMONSEARCH` to be `COMMONSEARCH` and changing the file name of `DEFAULTIPNODES` to be `USER1.ETC.IPNODES`.

```
f resolver,refresh,setup=user55.ressetup

EZZ9298I DEFAULTTCPIPDATA - None
EZZ9298I GLOBALTCPIPDATA - SYS1.TCPPARMS(TCPDATA)
EZZ9298I DEFAULTIPNODES - USER1.ETC.IPNODES
EZZ9298I GLOBALIPNODES - None
EZZ9304I COMMONSEARCH
EZZ9304I CACHE
EZZ9298I CACHESIZE - 200M
EZZ9298I MAXTTL - 2147483647
EZZ9298I UNRESPONSIVETHRESHOLD - 25
EZZ9293I REFRESH COMMAND PROCESSED
```

The following example contains the command and messages that are returned to delete all of the entries in the resolver cache.

```
f resolver,flush,all

EZZ9305I 200 CACHE ENTRIES DELETED
EZZ9293I FLUSH COMMAND PROCESSED
```

Usage

See the resolver setup statements in the resolver setup file topic in z/OS Communications Server: IP Configuration Guide for an explanation of the fields on the report.

MODIFY command: REXEC

Use the MODIFY command to change the parameters on the Remote Execution server.

Format

```
►►──┬─MODIFY─┬──procname──,──┬────────────────┬──,──┬────────────────┬──,──────────►
    └─F──────┘               └─EXIT=──exitmod─┘     └─TSOPROC=──proc─┘

►──┬──────────────┬──,──┬─────────────────┬──,──┬──────────────┬──,──────────────────►
   └─MSGCLASS=──c─┘     └─PURGE=──┬─Yes─┬─┘     └─TSCLASS=──c─┘
                                  └─No──┘

►──┬─────────────────────────────────┬──────────────────────────────────────────────►◄
   └─TRACE=──┬─LOG──────────────┬─┘
             ├─NOLOG────────────┤
             ├─SEND─────────────┤
             ├─NOSEND───────────┤
             ├─CLIENT=──client──┤
             ├─ALLCLIENTS───────┤
             └──┬───────┬───────┘
                └─RESET─┘
```

Parameters

For a description of the valid resolver setup statements parameters, see Remote execution server parameters in the z/OS Communications Server: IP Configuration Reference.

Examples

To change the user exit and TSO batch procedure, you might enter:

```
F RXSERVE,EXIT=USERX22,TSOPROC=KHFLACCN
```

Usage

You cannot use the MODIFY command to change the MAXCONN parameter.

MODIFY command: RPCBIND

Use the MODIFY command to start and stop tracing after the rpcbind address space initialization is complete.

Format

```
►►──MODIFY──jobname──,──TRACE──=──┬─FLOW───┬──────────────────────────────►◄
                                  ├─NOFLOW─┤
                                  ├─LOG────┤
                                  ├─NOLOG──┤
                                  ├─ON─────┤
                                  ├─OFF────┤
                                  ├─XDR────┤
                                  ├─NOXDR──┤
                                  └─?──────┘
```

Parameters

TRACE
Subcommand to begin general tracing. Tracing options include the following values:

FLOW
Enable tracing for entry and exit of modules.

NOFLOW
Disable tracing for entry and exit of modules.

LOG
Enable activity tracing for each RPC procedure on the server that was invoked by an RPC client.

NOLOG
Disable activity tracing for each RPC procedure on the server that was invoked by an RPC client.

ON Enable all tracing.

OFF
Disable all tracing.

XDR
Enable tracing of XDR procedures.

NOXDR
Disable tracing of XDR procedures.

? Display the trace status.

Result: Specifying TRACE on the MODIFY command is additive. Enabling tracing with the values FLOW and then XDR results in tracing for both. Specifying ON or OFF sets or resets all trace types.

MODIFY command: SMTP

The MODIFY SMTP command provides an interactive interface to the SMTP server that allows you to perform the following functions:

- Query the operating statistics of the SMTP server
- Query the SMTP mail delivery queues
- Perform privileged system administration tasks such as shutting down the SMTP server and enabling or disabling various tracing and debugging options

Format

```
►►──┬─MODIFY─┬──smtpprocname──,SMSG,──┬─DEbug──────────────────┬──────────────►◄
    └─F──────┘                        ├─EXpire,ipaddr──────────┤
                                      ├─HElp───────────────────┤
                                      ├─NODebug────────────────┤
                                      ├─NOTrace────────────────┤
                                      ├─NUMQueue───────────────┤
                                      │         ┌─,MAX=100───┐ │
                                      ├─QUeues──┼────────────┼─┤
                                      │         ├─,MAX=*─────┤ │
                                      │         └─,MAX=──lines─┘ │
                                      ├─SHutdown───────────────┤
                                      ├─STARTEXIT──────────────┤
                                      ├─STats──────────────────┤
                                      ├─STOPEXIT───────────────┤
                                      └─TRace──────────────────┘
```

Parameters

Tip: The minimum abbreviation for each parameter is shown in uppercase letters.

DEbug
Enables connection debugging and tracing, which sends information to the SMTP DEBUG data set. Specifying this parameter has the same effect as adding the DEBUG statement to the SMTP configuration data set (SMTPCONF).

EXpire,*ipaddr*
Causes the domain name resolution for mail queued for delivery to this IP address to expire. SMTP again attempts to resolve the IP addresses for this mail if the retry time interval has not expired.

HElp
Provides a list of valid SMTP SMSG commands.

NODebug
Disables connection debugging and tracing.

NOTrace
Disables resolver tracing.

NUMQueue
Provides the number of mail messages currently queued in SMTP.

QUeues
Provides a list of mail queued on the various SMTP mail processing queues.

SHutdown
Causes the SMTP server to shut down.

STARTEXIT
Causes SMTP to enable a user exit program by issuing the initialization call to the user exit program if one exists. For more information about user exit programs, see the z/OS Communications Server: IP Configuration Guide.

STats
Provides operating statistics about SMTP server events that have occurred since the SMTP server was started.

STOPEXIT
Causes SMTP to disable a currently running user exit program by issuing the

termination call to the user exit program if one exists. For more information about user exit programs, see the z/OS Communications Server: IP Configuration Guide.

TRace

Enables resolver tracing. The output of the resolver trace is sent to the SMTP console. The result is the same as adding TRACE RESOLVER to the TCPIP.DATA data set.

MAX

Limits the number of lines that are displayed to the MVS operator console for the QUeues report. Valid values are in the range 1 – 65 533. An asterisk (*) causes all output lines up to line 65 533 to be displayed. The default value is 100.

Examples

```
MODIFY SMTP,SMSG,DEBUG
EZA5597I SMSG DEBUG Output - Session Debugging Enabled

MODIFY SMTP,EXPIRE,123.123.123.123
EZA5598I SMSG EXPIRE Output - 123.123.123.123 - Mail queued for re-resolution

MODIFY SMTP,SMSG,HELP
EZA5593I SMSG HELP Output 376
Valid SMSG Commands:
QUeues,max=xxxx  - for mail queue lengths
NUMQueue - for total number of mail messages currently queued
STats    - for operating statistics
HElp     - to get this message
TRace    - to enable resolver tracing
NOTrace  - to disable resolver tracing
DEbug    - to enable session debugging
NODebug  - to disable session debugging
EXpire,a.b.c.d - to expire the domain name resolution for mail queued
                 for delivery to this IP address
SHutdown - to terminate the SMTP server
STARTEXIT- start/restart the user exit
STOPEXIT - stop the user exit


MODIFY SMTP,NODEBUG
EZA5599I SMSG NODEBUG Output - Session Debugging Disabled

MODIFY SMTP,NOTRACE
EZA5654I SMSG NOTRACE Output -  Resolver Tracing Disabled

MODIFY SMTP,SMSG,NUMQUEUE
EZA5596I SMSG NUMQUEUE Output -  Current Number of Mail Queued is 50

MODIFY SMTP,SMSG,QUEUE
EZA5594I SMSG QUEUE Output
----- Mail Queues -----
Spool Queue        : 0
R: xxx.xxx.xxx.xxx : 1 HostName.DomainName
Undeliverable Queue: 0
--- Resolver Queues ---
Process Queue:      0
Send Queue:         0
Wait Queue:         0
Retry Queue:        0
Completed Queue:    0
Error Queue:        0
```

Spool Queue

Contains mail that is destined for recipients on the local MVS system, or for recipients on an NJE system attached to the local MVS system. This queue is generally empty, because SMTP can deliver this mail quickly by spooling it to the local recipient or to the NJE address space for delivery to an NJE network recipient.

Active Indicates that if SMTP is currently transmitting to a TCP network destination, all the mail queued for that destination is shown to be active. Use the following format:

A: *xxx.xxx.xxx.xxx* : 1 *HostName.DomainName*

- *xxx.xxx.xxx.xxx* - The IP address
- 1 - The number of pieces of mail
- *HostName.DomainName* - The symbolic name

Queued

All mail that arrives over a batch SMTP connection, and mail from TCP connections that is to be forwarded to another TCP network destination through source routing, is placed on the queued list. As soon as SMTP receives resources from the TCP/IP address space, mail that is queued is considered to be active. The format is:

Q: *xxx.xxx.xxx.xxx* : 1 *HostName.DomainName*

- *xxx.xxx.xxx.xxx* - The IP address
- 1 - The number of pieces of mail
- *HostName.DomainName* - The symbolic name

Retry Queue

Mail is placed in this queue after SMTP attempts to transmit mail to each of the TCP network hosts but is unable to either open a connection or to complete delivery over the connection. After the number of minutes specified by the RETRYINT value, mail is promoted from the retry queue to the QUEUED state. For more information about the RETRYINT variable, see the z/OS Communications Server: IP Configuration Reference.

The format is:

R: *xxx.xxx.xxx.xxx* : 1 *HostName.DomainName*

- *xxx.xxx.xxx.xxx* - The IP address
- 1 - The number of pieces of mail
- *HostName.DomainName* - The symbolic name

Undeliverable Queue

Mail is placed in this queue if SMTP cannot deliver mail to a local MVS recipient or to a recipient on the NJE network attached to the local MVS system because spool space on the local MVS system is full.

After spool space has been increased and SMTP has been restarted, delivery attempts are resumed.

Resolver Queues

SMTP uses the following queues for processing queries to the name server. If the SMTP server is configured to use the site tables rather than the name server, these queues are not used. If the queue is empty, the word Empty appears to the right of the queue. If the queue contains queries, the queries appear on separate lines below the queue. However, because of the speed of the SMTP server, the output might indicate that the queue is active without containing any entries. In this case, the word Empty does not appear.

Process Queue
Contains queries waiting to be sent to the SMTP resolver. After the query has been processed, it is moved to the resolver send queue. This queue is typically empty.

Send Queue
Contains queries waiting to be processed by the SMTP resolver. SMTP staggers the number of queries sent by the resolver to prevent overloading the network and the name server.

Wait Queue
Contains queries for which the SMTP resolver is waiting for responses. Queries remain in this queue for the period of time it takes to receive a reply from the name server. If a reply is not received, the queries are removed from this queue after the resolver timeout has occurred, and are placed in the resolver retry queue. If the query is successful, the query is placed in the resolver completed queue.

Tip: The SMTP resolver timeout is specified by the RESOLVERTIMEOUT statement in the TCPIP.DATA data set.

Retry Queue
Contains queries that have previously failed, either because the name server did not reply, or the name server returned a temporary error that forced the SMTP resolver to retry the query. A temporary error occurs if, for example, the name server truncates a packet, or if the name server detects a processing error. The RESOLVERRETRYINT statement specifies the number of minutes SMTP waits before trying the query again. The RETRYAGE statement specifies the number of days SMTP continues to resolve the query before returning the mail to the sender.

Completed Queue
Contains queries that have been resolved and are waiting to be recorded into the mail. After the IP addresses are recorded, SMTP attempts to deliver the mail.

Error Queue
Contains queries that the name server has returned without answers. The corresponding mail message is returned to the sender with an unknown recipient error.

```
MODIFY SMTP,SHUTDOWN
EZA5655I SMSG SHUTDOWN Output - Stopping SMTP

MODIFY SMTP,STARTEXIT
EZA5656I SMSG STARTEXIT Output - Exit started

MODIFY SMTP,SMSG,STATS

EZA5595I SMSG STATS Output 618
Last Up Time:  Sat, 29 Jul 06 17:07:10 EST
Statistics  : 07/29
From TCP    :     0
From Spool  :   500
BSMTP Logs  :     0
Error Mail  :     0
To Local    :     0
To RSCS     :     0
To TCP      :   500
Passive Opns:     0
Active Opens:   400
```

```
-------------------------------
Highest num queued: 50
High reached at: Date: Sat, 29 Jul 06 17:07:09 EST
```

Last Up Time
The date and time that SMTP was last started.

Statistics
Statistics about mail handled by SMTP over the past four days including the following information:

From TCP
Number of pieces of mail that arrived over TCP connections

From Spool
Number of pieces of mail that arrived from spool (local or NJE senders)

BSMTP Logs
Number of pieces of mail generated in response to requests to VERBose batch SMTP connections

Error Mail
Number of pieces of mail generated to return error mail to the sender

To Local
Number of pieces of mail delivered to local recipients

To RSCS
Number of pieces of mail delivered to recipients on the RSCS network

To TCP
Number of pieces of mail delivered to recipients on the TCP/IP network

Passive Opns
Number of TCP connections through which mail was received

Active Opens
Number of TCP connections through which mail was delivered

Highest num queued
Highest number of messages queued in SMTP and the time and date this occurred

High reached at
Date and time that the Highest num queued value was reached

```
MODIFY SMTP,STOPEXIT
EZA5657I SMSG STOPEXIT Output - Exit Stopped
```

```
MODIFY SMTP,TRACE
EZA5658I SMSG TRACE Output - Resolver Tracing Enabled
```

MODIFY command: SNALINK LU0

Use the MODIFY command to halt the SNALINK LU0 interface.

Format

```
►►──┬─MODIFY─┬──procname──,──HALT──────────────────────────────────────────►◄
    └─F──────┘
```

Parameters

procname
The member name of the cataloged procedure used to start the SNALINK LU0 interface.

HALT
Shuts down the SNALINK interface.

MODIFY command: SNALINK LU 6.2

You can stop or restart the SNALINK LU6.2 interface and control tracing with the MODIFY command. Use the MODIFY command to:

- Stop or restart the SNALINK LU6.2 interface
- Alter the level of tracing

Format

```
►►──┬─MODIFY─┬──procname──,──┬──────────────────────────────────────┬──►◄
    └─F──────┘               ├─CANCEL───────────────────────────────┤
                             ├─DROP──┬─IP=──dest_ip─┬───────────────┤
                             │       ├─LU=──dest_lu─┤               │
                             │       └─ALL──────────┘               │
                             ├─HALT─────────────────────────────────┤
                             │       ┌─ACTIVE───────┐               │
                             ├─LIST──┼──────────────┼───────────────┤
                             │       ├─IP=──dest_ip─┤               │
                             │       ├─LU=──dest_lu─┤               │
                             │       └─ALL──────────┘               │
                             │          ┌─INIT─────────┐            │
                             ├─RESTART──┼──────────────┼────────────┤
                             │          ├─IP=──dest_ip─┤            │
                             │          ├─LU=──dest_lu─┤            │
                             │          └─ALL──────────┘            │
                             │        ┌─ON─────┐                    │
                             └─TRACE──┼────────┼──┬─IP=──dest_ip─┬──┘
                                      ├─OFF────┤  └─ALL──────────┘
                                      └─DETAIL─┘
```

Parameters

procname
The member name of the cataloged procedure used to start the SNALINK LU6.2 interface.

CANCEL
Cancels the SNALINK LU6.2 interface by a user abend. The system produces a dump and writes it to the data set defined by the //SYSUDUMP DD statement in the cataloged procedure.

DROP
Ends the connection with the destination nodes as specified.

IP=*dest_ip*
The destination IP address of the connection to end.

LU=*dest_lu*
The destination LU name of the connection to end. For dependent LU connections, either the sending or receiving remote LU name can be supplied and both sessions are ended.

ALL
Drops all connections defined in SNALINK LU6.2 configuration data set.

HALT
Shuts down the SNALINK LU6.2 interface.

LIST
Displays status and traffic information for the range of connections specified.

ACTIVE
The range of destinations to be listed. Information is displayed for all currently established connections handled by the specified address space. This is the default.

IP=*dest_ip*
The destination IP address of the connection to be listed.

LU=*dest_lu*
The destination LU name of the connection to be listed. For dependent LU connections, you can supply either the remote sending or receiving LU name.

ALL
Displays information for all destinations defined in the SNALINK LU6.2 configuration data set.

RESTART
Establishes one or more connections to destination nodes. Any destinations in the specified range that are already connected are skipped.

INIT
The range of connections to be established. If the INIT parameter is specified, connections are established with all destinations defined with the INIT parameter in the SNALINK LU6.2 configuration data set. If the RESTART subcommand is entered without parameters, the INIT option is the default.

IP=*dest_ip*
The destination IP address of the connection to be established.

LU=*dest_lu*
The destination LU name of the connection to be established. For dependent LU connections, either the remote sending or receiving LU name can be supplied and both sessions are established.

ALL
The range of connections to be established. If the ALL parameter is specified, connections are established with all destinations defined in the SNALINK LU6.2 configuration data set.

TRACE
Alters the levels of trace defined in the SNALINK LU6.2 configuration data set while the address space is active.

ON
Enables a basic level of tracing for all connection in the specified range. The default is ON.

OFF
If the OFF parameter is specified, tracing is disabled for all connections in the specified range.

DETAIL
Enables a detailed level of tracing for all connections in the specified range.

IP=*dest_ip*
The destination IP address associated with the connection for which tracing will be enabled or disabled.

ALL If the ALL parameter is specified, tracing for all destinations (either currently or subsequently connected) is set to the requested level.

Examples

To enable tracing for the procedure LU62PROD on connection associated with 9.163.37.12, enter

```
F LU62PROD, TRACE IP=9.163.37.112
```

The following example illustrates the output you might get if you issued the MODIFY command with the LIST parameter:

```
MODIFY TCPIPL62,LIST ALL

    TCPL62217I LIST Accepted; Range = All Connections
    TCPL62212I   192.9.207.39 (Connected on 92.013 at 09:52:11)
    TCPL62213I     Connected by:  DATA             Trace Level: OFF
    TCPL62214I     SEND:-  Status: Not Allocated    Packets Out: 0
    TCPL62215I     RECV:-  Status: Allocated        Packets In:  0
    TCPL62211I   192.9.207.40 (Disconnected on 92.013 at 08:30:10)
    TCPL62210I   192.9.207.41 (Disconnected)
    TCPL62219I LIST Completed
```

Usage

Determining the DLC connection status using NETSTAT DEVLINKS

For the SNALINK LU6.2 interface, the connection and disconnection of DLC links between the TCP/IP and SNALINK LU6.2 address spaces is independent of the connection and disconnection of VTAM links with destination nodes.

You can use the TSO command, NETSTAT DEVLINKS, to determine the status of the DLC connections between the main TCP/IP address space and the SNALINK LU6.2 address spaces.

Status Reported
Description

Inactive
The DLC connection has not been started. You can start one of the DLC links between TCP/IP and SNALINK LU6.2 with the VARY START command.

Issued Connect
The TCP/IP address space has issued a DLC connection request, but the SNALINK LU6.2 address space has not yet accepted the connection.

Connected
A DLC connection has been successfully established between the TCP/IP address space and the SNALINK LU6.2 address space.

Sending Message
A DLC connection has been successfully established between the 2 address

spaces, and a message has been sent by the TCP/IP address space, but it has not yet been received by the SNALINK LU6.2 address space.

Will retry connect
Either a previously connected DLC connection has been severed, or the previous connection request was not accepted within the timeout period. In either case, the TCP/IP address space attempts to resend another connection request within 30 seconds.

Status Reported
Explanation

`Issued connect`
Passive side: SNALINK is waiting for a remote LU to establish a session.

Active side: SNALINK is trying to establish a session with a remote LU.

`Will retry connect`
The last session was ended, or the last session attempt failed. SNAIUCV driver tries the connection again within 30 seconds.

`Connected`
An SNA send session is established. Under normal conditions this also means a receive session is established or will be established soon, and communication between the two LUs is possible.

`Sending message`
An SNA send session is established, and there is a DLC SEND currently outstanding.

MODIFY command: SNMP agent

Some SNMP agent initialization parameters can be modified while the agent is executing using the MVS MODIFY command. The MODIFY command can also be used to display the current level of SNMP agent tracing.

Format

```
►►──┬─MODIFY─┬──snmp_agent_jobname,──┬─INTERVAL=n──────────────┬──────────────────────►◄
    └─F──────┘                       └─TRACE,──┬─LEVEL=n─┬─┘
                                               └─QUERY───┘
```

Parameters

`INTERVAL`
Specifies an integer in the range 0 – 10, which indicates the maximum number of minutes before committed configuration changes to the SNMPD.CONF file will be written out. A value of 0 means that the changes will be written out at the time the SET is committed.

`TRACE`
Indicates SNMP agent tracing is to be queried or changed.

`LEVEL`
Specifies an integer in the range 0 – 255, which indicates the level of agent tracing. This corresponds to the -d parameter at agent initialization. See OSNMPD parameters in the z/OS Communications Server: IP Configuration Reference for additional guidance on setting the trace level.

QUERY
Requests that the current level of SNMP agent tracing be displayed.

MODIFY command: SNMP Network SLAPM2 subagent

You can control the Network SLAPM2 subagent (nslapm2) functions from the operator console using the MODIFY command. The following list shows the syntax and valid parameters.

Format

```
►►──┬─MODIFY─┬──procname──,──┬─Debug,Level=──n─┬──────────────────────────►◄
    └─F──────┘               ├─Cache,Time=──t──┤
                             └─Query───────────┘
```

Parameters

Debug,Level
Changes the Network SLAPM2 subagent start option debug level. *n* is the required debug level. Specifying a level of 0 disables debug tracing. If *n* is not specified, then the current debug level remains the same. See the Starting the network SLAPM2 subagent from the z/OS shell information in the z/OS Communications Server: IP Configuration Reference and the Problems connecting subagents to the SNMP agent information in the z/OS Communications Server: IP Diagnosis Guide for details about valid Network SLAPM2 subagent debug levels.

Cache,Time
Changes the Network SLAPM2 subagent start option cache time. *t* is the required cache time in seconds. If *t* is not specified, then the current cache time remains the same. See the Starting the network SLAPM2 subagent from the z/OS shell information in the z/OS Communications Server: IP Configuration Reference for details about valid Network SLAPM2 subagent cache times.

Query
Displays the current Network SLAPM2 subagent debug level, subagent cacheTime and actual cache time in effect.

MODIFY command: Syslog Daemon

Use the MODIFY command to control the syslog daemon functions from the operator console.

Format

```
►►──┬─MODIFY─┬──procname,──┬─ARCHIVE─────────────────────────────────────┬──►◄
    └─F──────┘             ├─DISPLAY,ARCHIVE──┬───────────────────────┬──┤
                           │                  │          ┌─,MAX=5─┐   │  │
                           │                  └─,DETAIL──┼────────┼───┘  │
                           │                             ├─,MAX=n─┤      │
                           │                             └─,MAX=*─┘      │
                           └─RESTART─────────────────────────────────────┘
```

Parameters

procname
The member name of the cataloged procedure that is used to start the syslog daemon.

ARCHIVE
Perform an immediate archive of UNIX file-system files that are defined by syslogd configuration rules. When the rule for the destination file includes the **-N** parameter and is preceded by the BeginArchiveParms statement, that file is archived. When the rule for the destination file includes the **-X** parameter, the contents of that file are deleted.

If a previously initiated archive is already in progress, an additional archive does not occur, and message FSUM1256 is issued to the console. The previously initiated archive might have been initiated by an ARCHIVE command, a time-of-day-based archive, or a file-system threshold-based archive.

DISPLAY,ARCHIVE
Display UNIX file-system use data for file systems that have output destination files defined by syslogd configuration rules. Capacity percentages are displayed for each file system. If you specify the DETAIL parameter, the largest files for that file system are displayed. The MAX parameter controls the number of files that are displayed for the detail report. The default value for MAX is 5; the maximum value is 65 535. If you specify an asterisk (*) then all files are displayed.

RESTART
Indicates that the syslogd configuration file should be reread. The result is similar to sending a SIGHUP signal to syslogd. Syslogd attempts to finish writing all pending output to the appropriate destinations before rereading the configuration file. If any output cannot be written within 30 seconds, the appropriate destination is marked as unreachable and any output pending for that destination is discarded.

If a previously initiated restart is already in progress, an additional restart is not performed, and message FSUM1256 is issued to the console. The previously initiated restart might have been initiated by a RESTART command or a SIGHUP signal.

Examples

The following command causes syslogd to reread its configuration file.

```
F SYSLOGD,RESTART
FSUM1254 SYSLOGD MODIFY COMMAND ACCEPTED
FSUM1252 SYSLOGD RECONFIGURATION COMPLETE
```

The following command causes syslogd to archive all UNIX file-system files that are defined by syslogd configuration rules.

```
F SYSLOGD,ARCHIVE
FSUM1254 SYSLOGD MODIFY COMMAND ACCEPTED
FSUM1260 SYSLOGD ARCHIVE COMPLETE FOR 2 FILES
```

The following commands cause syslogd to display UNIX file system utilization data.

```
F SYSLOGD,DISPLAY,ARCHIVE

FSUM1267 FILE SYSTEM SUMMARY 387
NAME=OMVS.VAR.HFS
```

```
PATH=/SYSTEM/var
512-BLOCKS=    177120 USED=    108592 AVAIL=     68528 USAGE= 61%
NAME=OMVS.VAR.LOGS.HFS
PATH=/SYSTEM/var/logs
512-BLOCKS=     60480 USED=     60096 AVAIL=       384 USAGE= 99%
```

F procname,DISPLAY,ARCHIVE,DETAIL

```
FSUM1268 FILE SYSTEM DETAILS 390
NAME=OMVS.VAR.HFS
PATH=/SYSTEM/var
512-BLOCKS=    177120 USED=    108592 AVAIL=     68528 USAGE= 61%
   FILE SIZE USAGE ARCHIVE PATH
       21214   12% NONE    /var/logu/daemon/daemon.trace
       20359   11% NONE    /var/logu/daemon/daemon.log
       13133    7% SEQ     /var/logu/all/all.logseq
       13133    7% SEQ     /var/temp/tempfull.log
       12900    7% SEQ     /var/logu/pagent/pagent.logseq
5 OF 24 RECORDS DISPLAYED
NAME=OMVS.VAR.LOGS.HFS
PATH=/SYSTEM/var/logs
512-BLOCKS=     60480 USED=     60096 AVAIL=       384 USAGE= 99%
```

F procname,DISPLAY,ARCHIVE,DETAIL,MAX=2

```
FSUM1268 FILE SYSTEM DETAILS 393
NAME=OMVS.VAR.HFS
PATH=/SYSTEM/var
512-BLOCKS=    177120 USED=    111664 AVAIL=     65456 USAGE= 63%
   FILE SIZE USAGE ARCHIVE PATH
       21218   12% NONE    /var/logu/daemon/daemon.trace
       20363   11% NONE    /var/logu/daemon/daemon.log
2 OF 24 RECORDS DISPLAYED
NAME=OMVS.VAR.LOGS.HFS
PATH=/SYSTEM/var/logs
512-BLOCKS=     60480 USED=     60096 AVAIL=       384 USAGE= 99%
```

MODIFY command: Trap forwarder daemon (TRAPFWD)

You can control the TRAPFWD daemon from the operations console using the MODIFY command. The following list shows the syntax and valid parameters.

Format

```
►►──┬─MODIFY─┬──trap_daemon_jobname,──┬─REFRESH──────────────┬──────────────────────►◄
    └─F──────┘                        └─TRACE,──┬─QUERY───┬──┘
                                                └─LEVEL=n─┘
```

Parameters

REFRESH
Dynamically refreshes the configuration information. When this is done, the old configuration information is discarded, the configuration file is read again, and the daemon is initialized.

TRACE
Indicates TRAPFWD tracing is to be queried or changed.

QUERY
Requests that the current level or TRAPFWD daemon tracing be displayed.

LEVEL
Valid values are:

- 0–No tracing.
- 1–Minimal tracing. Trace address from which the trap is received.
- 2–In addition to 1, trace addresses to which the trap packet is forwarded.

MODIFY command: VMCF and TNF

Display the names of current users of VMCF and TNF and remove names from the name list.

Format

```
►►──┬─MODIFY─┬──┬─VMCF,─┬──┬─DISPLAY,─┬──NAME=──┬─name─┬────────────────────►◄
    └─F──────┘  └─TNF,──┘  └─REMOVE,──┘         └─*────┘
```

Parameters

VMCF
Communicates with the VMCF address space.

TNF
Communicates with the TNF address space.

Display
Displays the current users of TNF/VMCF.

REMOVE
Terminates the current users of TNF/VMCF.

NAME
Named users or *=all users of the TNF/VMCF.

MODIFY command: X.25 NPSI server

Use the MODIFY command to pass parameters to the X.25 NPSI server.

Format

```
►►──┬─MODIFY─┬──procname──,──┬─CANCEL───────────────────┬─────────────────►◄
    └─F──────┘               ├─DEBUG ──digits───────────┤
                             ├─EVENTS ──┬────┬──────────┤
                             │          └─id─┘          │
                             ├─HALT─────────────────────┤
                             ├─LIST─────────────────────┤
                             ├─RESTART ──┬───────┬──────┤
                             │           └─mchlu─┘      │
                             ├─SNAP ──┬────┬────────────┤
                             │        └─id─┘            │
                             ├─TRACE ──┬─id─┬──┬─DATA─┬─┤
                             │         └─*──┘  └─OFF──┘ │
                             └─TRAFFIC──────────────────┘
```

Parameters

procname
The member name of the cataloged procedure used to start this server.

CANCEL
Cancels the X.25 NPSI server task and produces a dump.

DEBUG *digits*
Alters debug settings, where *digits* is a string of debug levels corresponding to those in the configuration data set for X.25 NPSI server.

EVENTS *id*
Displays event handler names for debugging, where *id* is an optional LU name or logon ID.

HALT
Shuts down the X.25 NPSI task, closing all connections.

LIST
Displays a list of the status of the virtual circuit.

RESTART *mchlu*
Attempts to reacquire failed links (MCHs), after reactivating them through VTAM. *mchlu* is an optional LU name from a link definition. If omitted, all inactive MCHs are restarted.

SNAP *id*
Displays program data areas for debugging, where *id* is an optional LU name or logon ID.

TRACE
Alters the trace level, where *id* is an optional LU name, logon ID, or an asterisk (*). TRACE can be one of two levels: DATA or OFF.

TRAFFIC
Displays traffic counts.

Examples

To halt an X.25 NPSI server whose procedure started with the following statements in the *hlq*.PROFILE.TCPIP you could issue either of the commands that follow at the operator console:

```
AUTOLOG
   TCPIPX25
```

Issue one of the following commands:

```
MODIFY TCPIPX25,HALT

F TCPIPX25,HALT
```

MODIFY command: z/OS Load Balancing Advisor

You can control the z/OS Load Balancing Advisor from the operator console using the MODIFY command.

Format

```
►►──┬─MODIFY─┬──procname,────────────────────────────────────────────►
    └─F──────┘
```

```
►──┬─DEBug,Level=debuglevel─────────────────────────┬────────────────►◄
   └─DISplay,──┬─DEBug──────────────────────────────┤
               │                    ┌─,MAX=100───┐  │
               └─LB──┬────────────────┬──┼────────────┼──┘
                     ├─,Index=lbindex─┤  ├─,MAX=*─────┤
                     └─,Index=ALL─────┘  └─,MAX=recs──┘
```

Parameters

procname
  The member name of the cataloged procedure used to start the z/OS Load Balancing Advisor.

DEBug,Level=*debuglevel*
  Changes the Advisor debug level. The needed debug level is *debuglevel*. See Debug settings and corresponding syslogd priority levels in the z/OS Communications Server: IP Diagnosis Guide and the Advisor debug_level statement in the z/OS Communications Server: IP Configuration Reference for details about valid Advisor debug levels.

DISplay,DEBug
  Displays the debug level in effect for the Advisor.

DISplay,LB
  Displays a summary of connected load balancers. The universally unique identifier (UUID), health value, flags, and an index are shown for each connected load balancer. The index will remain the same as long as the load balancer is connected.

DISplay,LB,MAX=*recs*
  Displays a summary of connected load balancers. The number of records (load balancers) displayed is limited by the MAX=*recs* parameter. The default is 100. If MAX=* is specified, then all connected load balancers are displayed.

DISplay,LB,Index=*lbindex*
  Displays all registered groups including detailed member data for the identified load balancer or for all connected load balancers (by specifying the ALL parameter). The *lbindex*value is the decimal index shown in the display of all load balancers. If you specify the ALL parameter, detailed member data for all connected load balancers is displayed.

DISplay,LB,Index=*lbindex***,MAX=***recs*
  Displays all registered groups including detailed member data for the identified load balancer or for all connected load balancers (by specifying the ALL parameter). The *lbindex* is the decimal index shown in the display of all load balancers. If you specify the ALL parameter, detailed member data for all connected load balancers is displayed. The number of records (members) displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, then all members are displayed.

Example 1 — The modify display LB command summarizes all load balancers that have connected to the Advisor.

```
F LBADV,DISP,LB
EZD1242I LOAD BALANCER SUMMARY
LB INDEX      : 00       UUID      : 637FFF175C
 IPADDR..PORT : 10.42.105.154..50005
 HEALTH       : 20       FLAGS     : NOCHANGE PUSH TRUST
```

```
LB INDEX      : 01        UUID      : 207FFF175C
 IPADDR..PORT : 10.42.105.60..50006
 HEALTH       : 7F        FLAGS     : PUSH TRUST
2 OF 2 RECORDS DISPLAYED
```

LB INDEX
Reference number used solely as the *lbindex* value on the MODIFY,DISPLAY,LB,INDEX= command. The same reference number is used for a load balancer as long as it is connected.

UUID
A hexadecimal value of the universally unique identifier assigned by the load balancer. This byte array can be up to 64 bytes in length.

IPADDR..PORT
The remote IP address and port at which the Advisor is connected to this load balancer. The IP address can be an IPv4 or an IPv6 address.

HEALTH
A hexadecimal value supplied by the load balancer that indicates the general health of the load balancer. Valid values are in the range 0 – X'7F'.

FLAGS
Flags that are set are displayed. Flag values are:

NOCHANGE
The Advisor sends only weights that have changed to the load balancer.

PUSH
The Advisor sends weights to the load balancer when any weights change.

TRUST
The load balancer trusts member applications to register themselves. Ignored by the Advisor.

Example 2 — The modify display command supplies details about a specific load balancer. The load balancer is identified using the index shown in the output of the modify display LB command. For each group of target applications, the display shows each active registered instance of the group in the sysplex.

```
F LBADV,DISP,LB,I=0
EZD1243I LOAD BALANCER DETAILS
LB INDEX      : 00        UUID      : 637FFF175C
 IPADDR..PORT : 10.42.105.154..50005
 HEALTH       : 20        FLAGS     : NOCHANGE PUSH TRUST
 GROUP NAME   : SYSTEMFARM
  GROUP FLAGS : BASEWLM
  IPADDR..PORT: 10.42.154.105..0
   SYSTEM NAME: MVS209    PROTOCOL  : 000  AVAIL     : YES
   WLM WEIGHT : 00040     CS WEIGHT : 100  NET WEIGHT: 00001
     Raw           CP: 40  zAAP: 60  zIIP: 00
     Proportional  CP: 40  zAAP: 00  zIIP: 00
   FLAGS      :
  IPADDR..PORT: 10.42.105.60..0
   SYSTEM NAME: VIC007    PROTOCOL  : 000  AVAIL     : YES
   WLM WEIGHT : 00050     CS WEIGHT : 100  NET WEIGHT: 00001
     Raw           CP: 50  zAAP: 00  zIIP: 00
     Proportional  CP: 00  zAAP: 00  zIIP: 00
      FLAGS      :
  IPADDR..PORT: 10.42.105.22..0
   SYSTEM NAME: N/A       PROTOCOL  : 000  AVAIL     : NO
   WLM WEIGHT : 00000     CS WEIGHT : 000  NET WEIGHT: 00000
     Raw           CP: 00  zAAP: 00  zIIP: 00
     Proportional  CP: 00  zAAP: 00  zIIP: 00
   FLAGS      : NOTARGETSYS
  IPADDR..PORT: 10:1::4:5..0
```

```
   SYSTEM NAME: MVS209    PROTOCOL  : 000  AVAIL     : NO
   WLM WEIGHT : 00040     CS WEIGHT : 000  NET WEIGHT: 00000
     Raw           CP: 40  zAAP: 60  zIIP: 00
     Proportional  CP: 40  zAAP: 00  zIIP: 00
   FLAGS      : NOTARGETIP
 GROUP NAME   : UDP_SERVER_FARM
  GROUP FLAGS : SERVERWLM
  IPADDR..PORT: 10.42.154.105..7777
   SYSTEM NAME: MVS209    PROTOCOL  : UDP  AVAIL     : YES
   WLM WEIGHT : 00021     CS WEIGHT : 100  NET WEIGHT: 00001
     Raw           CP: 20  zAAP: 22  zIIP: 00
     Proportional  CP: 10  zAAP: 11  zIIP: 00
   ABNORM     : 00200     HEALTH    : 100
   FLAGS      :
  IPADDR..PORT: 2001:DB8::10:5:6:2..7777
   SYSTEM NAME: MVS209    PROTOCOL  : UDP  AVAIL     : YES
   WLM WEIGHT : 00021     CS WEIGHT : 100  NET WEIGHT: 00001
     Raw           CP: 25  zAAP: 18  zIIP: 00
     Proportional  CP: 10  zAAP: 11  zIIP: 00
   FLAGS      :
  IPADDR..PORT: 10.42.105.60..7777
   SYSTEM NAME: VIC007    PROTOCOL  : UDP  AVAIL     : YES
   WLM WEIGHT : 00045     CS WEIGHT : 100  NET WEIGHT: 00002
     Raw           CP: 50  zAAP: 18  zIIP: 00
     Proportional  CP: 30  zAAP: 15  zIIP: 00
   FLAGS      :
 GROUP NAME   : CICS_SERVER_FARM
  GROUP FLAGS : BASEWLM
  ProcType    :
    CP : 60  zAAP: 40  zIIP: 00
  IPADDR..PORT: 10.42.154.105..8888
   SYSTEM NAME: MVS209    PROTOCOL  : TCP  AVAIL     : YES
   WLM WEIGHT : 00048     CS WEIGHT : 100  NET WEIGHT: 00001
     Raw           CP: 40  zAAP: 60  zIIP: 00
     Proportional  CP: 24  zAAP: 24  zIIP: 00
   FLAGS      :
  IPADDR..PORT: 10.42.105.60..8888
   SYSTEM NAME: VIC007    PROTOCOL  : TCP  AVAIL     : YES
   WLM WEIGHT : 00054     CS WEIGHT : 100  NET WEIGHT: 00001
     Raw           CP: 50  zAAP: 60  zIIP: 00
     Proportional  CP: 30  zAAP: 24  zIIP: 00
   FLAGS
  IPADDR..PORT: 10.42.105.22..8888
   SYSTEM NAME: N/A       PROTOCOL  : TCP  AVAIL     : NO
   WLM WEIGHT : 00000     CS WEIGHT : 000  NET WEIGHT: 00000
     Raw           CP: 00  zAAP: 00  zIIP: 00
     Proportional  CP: 00  zAAP: 00  zIIP: 00
   FLAGS      : NOTARGETSYS
  IPADDR..PORT: 10:1::4:5..8888
   SYSTEM NAME: MVS209    PROTOCOL  : TCP  AVAIL     : NO
   WLM WEIGHT : 00048     CS WEIGHT : 000  NET WEIGHT: 00001
     Raw           CP: 40  zAAP: 60  zIIP: 00
     Proportional  CP: 24  zAAP: 24  zIIP: 00
   FLAGS      : NOTARGETIP
7 OF 7 RECORDS DISPLAYED
```

For explanations of **LB INDEX, UUID, IPADDR..PORT, HEALTH, and FLAGS** see Example 1.

GROUP

The name of a group of related target applications. The group name is a UTF-8 string displayed in EBCDIC on the MVS console. Any non-displayable character is displayed as a question mark (?).

GROUP FLAGS

Flags that are currently applied to the group as a whole. Flag values are:

BASEWLM
Indicates that system WLM recommendations are being used to calculate the net weight for each member of the group.

BASEWLM*
Indicates that SERVERWLM was coded on the Advisor WLM statement or was specified for this group on the PORT_LIST Advisor statement in order to use server-specific WLM recommendations. However, the Advisor is using system WLM recommendations instead to calculate the net weight for each member of the group. This can occur if one or more of the Agents owning the members within the group does not support server-specific WLM recommendations.

SERVERWLM
Indicates that server-specific WLM recommendations are being used to calculate the net weight for each member of this group.

proctype
When BASEWLM recommendations are configured, the *proctype* value indicates the expected proportion of each type of processor that a target application's workloads will consume. A composite recommendation is determined from these proportions. A PROCTYPE value can be configured on the port_list or wlm statement; when this value is not configured, it assumes a default value to indicate that the composite recommendations include only the general CP weight.

CP The expected general CP utilization proportion that will be consumed by the applications.

zAAP
The expected zAAP utilization proportion that will be consumed by the applications.

zIIP
The expected IBM System z9® or later Integrated Information Processor (zIIP) utilization proportion that will be consumed by the applications.

Restrictions:
- zAAP and zIIP weight recommendations are available only if all systems in the sysplex are z/OS release V1R9 or later. If all systems in the sysplex are not z/OS release V1R9 or later, only CP weights are considered when determining a composite weight recommendation.
- zAAP and zIIP weight recommendations are not used when determining the composite weight for system members.

IPADDR..PORT
Indicates the IP address and port to which the target application is bound. This is the first of several lines relating to the same target application. If this represents a system member, then IPADDR represents an IP address belonging to a TCP/IP stack on one of the MVS systems in the sysplex, and the PORT will be 0.

SYSTEM NAME
Indicates the MVS system name of the MVS system where the application exists. If this is a system member, this indicates the MVS system name of the MVS system that owns the IP address.

PROTOCOL
Indicates the protocol that the application is using. If the protocol is not TCP or UDP, the decimal number of the protocol is displayed. For system members, this will be 0.

AVAIL
Indicates whether the member is available for new workload distribution. A value of **YES** indicates that the Advisor considers the application available for load balancing. A value of **NO** indicates that the Advisor recommends that the application not be considered for load balancing.

WLM WEIGHT
Indicates the Workload Manager weight value for the MVS system or the server-specific WLM weight based on the BASEWLM or SERVERWLM group flag. This value is in the range 0 – 64. This value is the composite weight; it is the sum of the displayed proportional CP, zAAP, and zIIP weights for this member.

CP When the distribution method is BASEWLM the following apply:

- The Raw value is the WLM system general CP weight recommendation. The value is based on the amount of displaceable general CPU capacity on this system as compared to the other target systems.
- The Proportional value is the Raw value modified by the expected general CP utilization proportion configured on the portlist and wlm statement for this application.

When the distribution method is SERVERWLM the following apply:

- The Raw value is the WLM server-specific general CP recommendation. This is the amount of displaceable general CPU capacity based on the application workload's importance (as defined by the WLM policy) as compared to the other target systems.
- The Proportional value is the Raw value modified by the proportion of general CP capacity that is currently being consumed by the application's workload as compared to the other processors (zAAP and zIIP).

zAAP
When the distribution method is BASEWLM the following apply:

- The Raw value is the WLM system zAAP weight recommendation. This value is based on the amount of displaceable zAAP capacity on this system as compared to the other target systems.
- The Proportional value is the Raw value modified by the expected zAAP utilization proportion configured on the portlist and wlm statement for this application.

When the distribution method is SERVERWLM the following apply:

- The Raw value is the WLM server-specific zAAP recommendation. This value is the amount of displaceable zAAP capacity based on the application workload's importance (as defined by the WLM policy) as compared to the other target systems.
- The Proportional value is the Raw value modified by the proportion of zAAP capacity that is currently being consumed by the application's workload as compared to the other processors (general CPU and zIIP).

zIIP
When the distribution method is BASEWLM the following apply:

- The Raw value is the WLM system zIIP weight recommendation. This value is based on the amount of displaceable zIIP capacity on this system as compared to the other target systems.
- The Proportional value is the Raw value modified by the expected zIIP utilization proportion configured on the portlist and wlm statements for this application.

When the distribution method is SERVERWLM the following apply:

- The Raw value is the WLM server-specific zIIP recommendation. This value is the amount of displaceable zIIP capacity based on the application workload's importance (as defined by the WLM policy) as compared to the other target systems.
- The Proportional value is the Raw value modified by the proportion of zIIP capacity that is currently being consumed by the application's workload as compared to the other processors (general CPU and zAAP)

Restrictions:

- zAAP and zIIP weight recommendations are available only if all systems in the sysplex are z/OS release V1R9 or later. If all systems in the sysplex are not z/OS release V1R9 or later, only CP weights are considered when determining a composite weight recommendation.
- zAAP and zIIP weight recommendations are not used when determining the composite weight for system members.

CS WEIGHT

Indicates the weight value recommended by the Agent. The range is 0 – 100, with a higher weight indicating that the application is able to handle more work than an application with a lower weight. One exception is that when the Agent is gathering historical data for an application (which takes 2 update intervals), the weight will be 100 and the NODATA flag will be set.

NET WEIGHT

Indicates the relative weight of this application in the sysplex. A higher weight indicates that an application can handle more workload than a lower weight application in the same group. This weight is based upon the WLM weight, the CS weight, the number of members in each group, and other factors. Net weights should be compared only within a group. Weights within a group are then normalized to yield the net weight. Normalization involves reducing the weight values while largely preserving the ratios between the weights. Normalization is performed within a group only if there is more than one available member in the group. Each group is calculated separately.

Result: In some cases, the value of NET WEIGHT is 1 (when the WLM WEIGHT or CS WEIGHT of all available members in the group is zero). This is done to force the load balancer to distribute workload in a round-robin fashion to those members rather than allowing the load balancer to potentially halt workload distribution to the entire group.

ABNORM

This field is displayed if the GROUP FLAGS values indicate that server-specific (SERVERWLM) WLM recommendations are being used. The value is nonzero if the server application is experiencing conditions in which transactions are completing abnormally. It represents a rate of abnormal transaction completions per 1000 total transaction completions. It is applicable only for target applications such as IWMRPT that act as Subsystem Work Managers and report transaction status using Workload Management Services. For example, the value 200 in this example indicates that 20% of all transactions processed by the server application are completing abnormally. Under normal conditions or if the server is not providing this information to WLM, this value should be 0.

A nonzero value indicates that the server application has reported some abnormal transaction completions to WLM and that WLM has reduced the server-specific recommendation for this server instance. The greater the value of this field, the greater the reduction in the recommendation provided by WLM. For more information about the conditions that cause the abnormal transaction completions for a given server application, see the documentation provided by the server application.

Restriction: Although WLM uses abnormal transaction completion rate information that is provided by the application to reduce the server-specific recommendation, this information is available for display on an Advisor only if the Load Balancing Agents and the Advisor are running on a z/OS V1R8 system or later. A z/OS V1R7 Load Balancing Agent does not provide this information to the Load Balancing Advisor. In this situation, a z/OS V1R8 Advisor shows a normal abnormal transaction completion rate of 0 even if WLM is reducing the server-specific recommendation because of a nonzero abnormal transaction completion rate reported from the application.

HEALTH

This field is displayed if the GROUP FLAGS values indicate that server-specific (SERVERWLM) WLM recommendations are being used. This health indicator is available only for applications that provide this information to WLM using the IWM4HLTH or IWMSRSRG services. It indicates the general health of an application or subsystem. Under normal circumstances or if the server is not providing this information to WLM, the value of this field is 100, indicating that the server is 100% healthy.

Values less than 100 indicate that the server is experiencing problem conditions that are not enabling it to process new work requests successfully; this causes WLM to reduce the server-specific recommendation for this server instance. The lower the value of this field, the greater the reduction in the recommendation provided by WLM.

Restriction: Although WLM uses the health indicator provided by the application to reduce the server-specific recommendation, this information is available for display on an Advisor only if the Load Balancing Agents and the Advisor are running on a z/OS V1R8 system or later. A z/OS V1R7 Load Balancing Agent does not provide this information to the Load Balancing Advisor. In this situation, a z/OS V1R8 Advisor shows a normal health indicator of 100 even if WLM is reducing the server-specific recommendation because of an abnormal health indication from the application.

FLAGS

Flag values that are currently set. Flag values are:

LBQ

Load Balancer quiesce, which means that the load balancer has requested that no more additional work be assigned to the quiesced application or system.

NOTARGETAPP

Indicates that an Agent found the IP address configured on a TCP/IP stack, but the Agent did not find a specific application using the same port and protocol.

NOTARGETIP
Indicates that an Agent found the IP address configured on a TCP/IP stack, but the address is not usable. For example, the IP address might be unavailable.

NOTARGETSYS
Indicates that no Agent found this IP address.

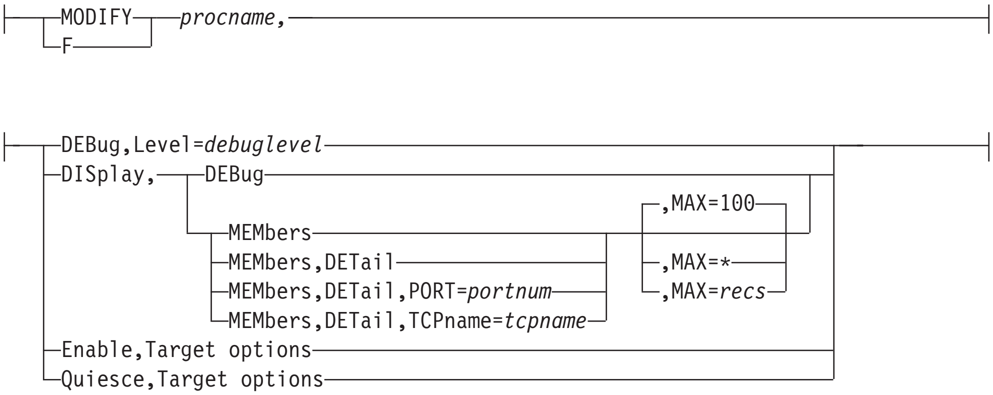
NODATA
Indicates that an Agent has reported this application but does not yet have the historical data to recommend a CS weight.

OPQ
Operator quiesce, which means that the MVS operator at the owning Agent has requested that no more additional work be assigned to the quiesced application or system.

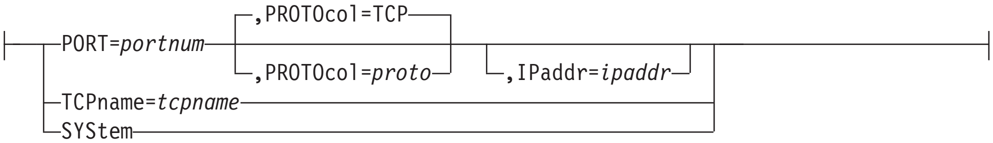
MODIFY command: z/OS Load Balancing Agent

You can control the z/OS Load Balancing Agent from the operator console using the MODIFY command.

Format



Target options:



Parameters

procname
The member name of the cataloged procedure used to start the Agent.

DEBug,Level=*debuglevel*
Changes the Agent debug level. The required debug level is *debuglevel*. See the debug_level statement description in the z/OS Communications Server: IP Configuration Reference and Debug settings and corresponding syslogd priority levels in the z/OS Communications Server: IP Diagnosis Guide for details about valid Agent debug levels.

DISplay,DEBbug
Displays the debug level in effect for the Agent.

DISplay,MEMbers
Displays a summary of information about all registered local applications and systems.

DISplay,MEMbers,DETail
Displays detailed information about all registered local applications and systems.

DISplay,MEMbers,DETail,PORT=*portnum*
Displays detailed information about all registered local applications that are bound to the specified port (or system members if PORT=0 is entered).

DISplay,MEMbers,DETail,TCPname=*tcpname*
Displays detailed information about all registered local applications or system members that are associated with the specified TCP/IP address space. The *tcpname* value must be less than or equal to 8 characters in length.

DISplay,MEMbers,...,MAX=*recs*
Displays member information according to the specified parameters. The number of records (members) displayed is limited by the MAX=*recs* parameter. The default value is 100. If MAX=* is specified, all members are displayed.

Enable
Mark all matching quiesced active registered applications or system members as enabled. The Agent advises the load balancer to route work to the target applications.

Quiesce
Mark all matching active registered applications or system members as quiesced. The Agent advises the load balancer not to route work to the target applications.

Target options:

Either PORT, TCPNAME, or SYSTEM is required for ENABLE and for QUIESCE.

PORT=*portnum***[,PROTOcol=***proto***][,IPaddr=***ipaddr***]**
Mark all active registered target applications or system members using the specified target port as enabled or quiesced. The port number is a decimal value. If more than one application is sharing a port, all the applications are enabled or quiesced. You can further identify the applications to be enabled or quiesced by specifying the TCP or UDP keyword or by specifying the decimal protocol number. TCP is the default. Therefore, if you specify a system member (PORT=0), you must also specify PROTOCOL=0. To uniquely specify one specific application, use the IPADDR option. The port number, protocol, and (optionally) IP address are ANDed.

TCPname=*tcpname*
Mark all active registered target applications and system members associated with this TCP/IP address space as enabled or quiesced. The *tcpname* value must be less than or equal to 8 characters in length.

SYStem
Mark all active registered target applications and system members on this system as enabled or quiesced.

Example — Display detailed information about all registered local applications and system members.

```
F LBAGENT,DISP,MEM,DET
EZD1245I MEMBER DETAILS
LB INDEX      : 00        UUID      : 637FFF175C
 GROUP NAME   : SYSTEMFARM
  IPADDR..PORT: 10.42.105.154..0
   TCPNAME    : TCPCS     MATCHES   : 001  PROTOCOL  : 000
   FLAGS      :
   JOBNAME    : N/A       ASID      : N/A  RESOURCE  : N/A
  IPADDR..PORT: 10:1::4:5..0
   TCPNAME    : TCPCS5    MATCHES   : 000  PROTOCOL  : 000
   FLAGS      :
   JOBNAME    : N/A       ASID      : N/A  RESOURCE  : N/A
 GROUP NAME   : UDP_SERVER_FARM
  IPADDR..PORT: 10.42.105.154..7777
   TCPNAME    : TCPCS     MATCHES   : 001  PROTOCOL  : UDP
   FLAGS      : ANY
   JOBNAME    : TESTD1    ASID      : 0035 RESOURCE  : 000000A3
  IPADDR..PORT: 2001:DB8::10:5:6:2..7777
   TCPNAME    : TCPCS2    MATCHES   : 001  PROTOCOL  : UDP
   FLAGS      : ANY V6
   JOBNAME    : TESTD2    ASID      : 002A RESOURCE  : 00000031
4 OF 4 RECORDS DISPLAYED
```

LB INDEX, UUID, GROUP NAME, and IPADDR..PORT
For explanations of these items, see Example 1.

TCPNAME
The name of the Communications Server stack that owns the IP address in this member.

MATCHES
The number of ports on which the application is running. For applications sharing a port, this value can be more than 1. If the value of matches is zero, the Agent found the member's IP address reported by an active TCP/IP stack, but did not find a target application or system. For additional debugging information, see the z/OS Communications Server: IP Diagnosis Guide.

PROTOCOL
The protocol that the target application is using. If the protocol is not TCP or UDP, the decimal number of the protocol is displayed.

FLAGS
The flags that are currently set. Flag values are:

ANY
Indicates that the application is bound to INADDR_ANY or the unspecified IPv6 address (in6addr_any).

DISC
Indicates that the Agent is disconnected from the Advisor. The Agent continues trying to connect to the Advisor. Data displayed when the DISC flag is shown is current TCP/IP data for the last set of targets that were received from the Advisor.

NODATA
Indicates that the Agent is temporarily reporting a Communications Server weight (CS Weight) of 100 for the application. Two update intervals are needed for weight calculation so that the Agent calculates the weight beginning at the second update interval. CS Weight might continue to be reported as 100 at this point if the server is healthy. Configure the update interval in the Advisor configuration file (see the debug_level statement description in the z/OS Communications Server: IP Configuration Reference for details).

SYSQ, TCPQ, or APPQ

Operator quiesce, which means that the operator has requested that no more additional work be assigned to the quiesced application or system member. The different flags reflect the highest level of quiesce command that applies, and also the type of enable command that must be used to enable the application or system member.

SYSQ

Indicates that the application or system member was quiesced with the `F procname,QUIESCE,SYSTEM` command, and that the `F procname,ENABLE,SYSTEM` command must be used to enable it.

TCPQ

Indicates that the application or system member was quiesced with the `F procname,QUIESCE,TCPNAME=`*tcpname* command, and that the `F procname,ENABLE,TCPNAME=`*tcpname* command must be used to enable it.

APPQ

Means that the application or system member was quiesced with the `F procname,QUIESCE,PORT=` *port* command, and the `F procname,ENABLE,PORT=` *port* command must be used to enable it.

V6 Indicates the IPv6_V6ONLY socket option. It is able to communicate only with IPv6 clients

JOBNAME

The MVS job name of the target application or system member.

Result: Displays as `N/A` for system members (port=0 and protocol=0).

ASID

The MVS address-space identifier of the target application or system member.

Result: Displays as `N/A` for system members (port=0 and protocol=0).

RESOURCE

An identifier that uniquely identifies one instance of an application or system member. If an application is stopped and started, the same job name and ASID could be reused, but with a different resource identifier. The resource identifier is also displayed in the DISPLAY TCPIP,,NETSTAT,CONN command.

Result: Displays as *N/A* for system members (port=0 and protocol=0).

VARY TCPIP command

Use the VARY TCPIP command from the MVS operator console to display help for a supported command or to control some functions of the address space that corresponds to the started procedure name that was specified on the command. The abbreviated version of the command is the letter V.

This is the general format of the VARY command:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──parameter──────────────────────────►◄
                     └─procname─┘
```

procname

The name of the member in a procedure library that was used to start the server or address space. You can omit the *procname* parameter when you direct the command to a TCP/IP stack address space and only one TCP/IP stack is currently active.

parameter
Any of the parameters that are valid for the server.

The following servers or address spaces support the MVS VARY TCPIP command. Not all servers support the same parameters. For further descriptions of the supported parameters, see Table 10.

Table 10. Servers or address spaces that support the MVS VARY TCPIP command

Server or address space	Main parameters	Additional information
TCP/IP address space	DATTRACE, DROP, OBEYFILE, OSAENTA, PKTTRACE, PURGECACHE, START, STOP, SYNTAXCHECK, SYSPLEX	See “VARY command: TCP/IP address space” on page 241
TN3270E Telnet server address space	HELP, OBEYFILE, TELNET, LUNS	See “VARY command: TN3270E Telnet server address space” on page 273

Security considerations for the VARY command

You can restrict access to the VARY TCPIP command by defining RACF® profiles under the OPERCMDS class and specifying the list of users that are authorized to issue the VARY TCPIP command. You can decide on the level of control that is appropriate for your installation. For example, you might want to allow a user to be able to start or stop a TCP/IP device using the VARY TCPIP command but you do not want the user to be able to modify the TCP/IP configuration.

The RACF profile names that restrict access to each of the VARY TCPIP commands are listed under each command's usage notes. You can use the control statements in the sample JCL job that is provided in SEZAINST(EZARACF) to define these profile names.

Requirement: CONTROL access to each profile is required to enable you to issue the VARY TCPIP command.

To restrict all of the VARY TCPIP commands, you can define a generic profile as follows:

```
RDEFINE OPERCMDS (MVS.VARY.TCPIP.**) UACC(NONE)
PERMIT MVS.VARY.TCPIP.** ACCESS(CONTROL) CLASS(OPERCMDS)
   ID(USER1)
```

In this example, only user ID USER1 is allowed to issue any VARY TCPIP operator commands. In another example, if you wanted to restrict usage of the VARY TCPIP,,OBEYFILE command to user ID USER2 you could make the following definitions:

```
RDEFINE OPERCMDS MVS.VARY.TCPIP.OBEYFILE UACC(NONE)
PERMIT MVS.VARY.TCPIP.OBEYFILE ACCESS(CONTROL)
   CLASS(OPERCMDS) ID(USER2)
```

Note: If you want to restrict the use of the VARY TCPIP,,OBEYFILE command, you must issue RDEFINE OPERCMDS for MVS.VARY.TCPIP and MVS.VARY.TCPIP.OBEYFILE, and issue a subsequent PERMIT defining the specified ID that will have an ACCESS of at least CONTROL for the OPERCMDS class.

The RACF OPERCMDS class must be activated for any of these profiles to take effect. You must also ensure that the appropriate RACF options are specified to enable you to define generic RACF profiles for these profiles. This can be accomplished by the following RACF commands:

```
SETR CLASSACT(OPERCMDS)
SETR GENERIC(OPERCMDS)
SETR GENCMD(OPERCMDS)
SETR RACLIST(OPERCMDS)
```

Before the profiles take effect, a refresh of these RACF profiles might be required. This can be accomplished by the following RACF commands:

```
SETR GENERIC(OPERCMDS) REFRESH
SETR RACLIST(OPERCMDS) REFRESH
```

VARY command: TCP/IP address space

The functions listed in Table 11 support the VARY TCPIP command when it is directed to a TCP/IP stack address space.

Table 11. Functions that support the VARY TCPIP command

Function	Command
DATTRACE	"VARY TCPIP,,DATTRACE"
DROP	"VARY TCPIP,,DROP" on page 243
OBEYFILE	"VARY TCPIP,,OBEYFILE" on page 246
OSAENTA	"VARY TCPIP,,OSAENTA" on page 247
PKTTRACE	"VARY TCPIP,,PKTTRACE" on page 257
PURGECACHE	"VARY TCPIP,,PURGECACHE" on page 262
START or STOP	"VARY TCPIP,,START or VARY TCPIP,,STOP" on page 263
SYNTAXCHECK	"VARY TCPIP,,SYNTAXCHECK" on page 264
SYSPLEX	"VARY TCPIP,,SYSPLEX" on page 265

VARY TCPIP,,DATTRACE

Use the VARY TCPIP,,DATTRACE command to trace socket data (transforms) into and out of the physical file structure (PFS). For TCP and UDP sockets, this command also creates the following records:

- A Start record with the API Data Flow Starts State field that indicates the first data sent or received by the application for the associated socket.
- An End record with the API Data Flow Ends State field that indicates that the socket is closed.

Format:

```
►►──Vary──TCPIP,──┬──────────┬──,──DATtrace──┬───────┬──┤ TRACE ├──────────────►◄
                  └─procname─┘               ├─,ON───┤
                                             └─,OFF──┘
```

TRACE:

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report

```
   ┌─FULL──────────────────────────┐    ┌─JOBNAME=*────────────┐
├──┼───────────────────────────────┼─,──┼──────────────────────┼─,──────────────►
   │              ┌─200──────────┐ │    └─JOBNAME=job_name─────┘
   └─ABBREV=──────┼──────────────┼─┘
                  └─abbrev_length┘

   ┌─IP=*──────────────────────────┐┌─PORTNum=*──────────────┐
►──┼───────────────────────────────┼┼────────────────────────┼──────────────────┤
   └─IP=──┬─ IPv4_address ─┬───────┘└─PORTNum=port_number────┘
          └─ IPv6_address ─┘
```

IPv4_address:

```
                     ┌─,SUBNet=255.255.255.255─┐
├──ipv4_address──────┼─────────────────────────┼──────────────────────────────┤
                     ├─,SUBNet=subnet_mask─────┤
                     └─/num_mask_bits──────────┘
```

IPv6_address:

```
                     ┌─/128──────────┐
├──ipv6_address──────┼───────────────┼────────────────────────────────────────┤
                     └─/prefixLength─┘
```

Parameters:

procname
The name of the member in a procedure library that was used to start the server or address space.

ON Turns on socket data tracing, clears all settings previously defined, and refreshes just the default settings.

OFF
Turns off socket data tracing.

ABBREV
Specifies that a truncated portion of the socket data is to be traced. You can specify a length in the range 0 – 65 535 or use the default value 200. The ABBREV parameter can be used to reduce the volume of data stored in the trace file.

FULL
Specifies that all of the socket data is to be traced

JOBNAME
Specifies the name of the application address space to be traced. The default (*) is for all jobs.

IP Specifies an IP address (either a 32-bit IPv4 address in dotted decimal notation, or a 128-bit IPv6 address in colon hexadecimal notation) that is compared with both the source and destination addresses of associated sockets. If either the source or destination address of a socket matches the specified IP address, the data is traced. If the IP option is omitted, or an asterisk (*) is specified, then all IP addresses are traced.

If an IPv6 address is specified, then an optional *prefixLength* (range 1 – 128) is allowed. IPv4 addresses and IPv4-mapped IPv6 addresses are treated as equivalent addresses. The default *prefixLength* is 128. If an IPv4 address is specified, then /*num_mask_bits* can be used. The *num_mask_bits* and SUBNET are mutually exclusive. An error message is displayed if both are coded.

Note: IP address selection is not recommended for use with DATTRACE.

PORTNUM

Specifies a port number that is compared with the source and destination port numbers of associated sockets. The port number must be an integer in the range 1 – 65 535. If either the source or the destination port matches the specified port number, the data is traced. If you omit the PORTNUM option or if you specify an asterisk (*), packets are not filtered based on source or destination ports.

Rule: Packets that use the RAW protocol type are not traced if you specify the PORTNUM option. The PORTNUM parameter applies only to the TCP and UDP protocols.

SUBNET

Specifies a subnet mask that applies to the host and network portions of the IP address specified on the `IP=`*`ipv4_address`* parameter. The subnet mask must be specified in dotted decimal notation and must be specified in conjunction with the `IP=`*`ipv4_address`* parameter. With an IPv4 address specified, the /*num_mask_bits* can be used. The *num_mask_bits* and SUBNET are mutually exclusive. An error message is displayed if both are coded.

Examples:

You can start data traces for all job names using the VARY command:

- IPv4 addressing: `v tcpip,,dat,jobname=*,ip=9.67.113.61/32`
- IPv6 addressing: `v tcpip,,dat,full,jobname=*,ip=C5::1:2:3:4/126`

You can use the Netstat CONFIG/-f command to display data traces. The following example shows a data trace for a single entry.

```
Data Trace Setting:
   Jobname: *                TrRecCnt: 0000000000    Length: FULL
   IpAddr:  *                SubNet: 255.255.255.255
   PortNum: *
```

The following example shows a data trace for multiple entries:

```
Data Trace Setting:
 JobName: *          TrRecCnt: 00000000  Length: FULL
 IpAddr/PrefixLen:  10.1.1.1/24
 PortNum: *

 JobName: *          TrRecCnt: 00000000  Length: FULL
 IpAddr/PrefixLen:  5555:4444::2222/128
 PortNum: *
```

Usage:

- Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.DATTRACE.

VARY TCPIP,,DROP

Use the VARY TCPIP,,DROP command to drop a single TCP connection or UDP socket, or to stop all established TCP connections for a specified server. For detailed information about drop processing, see “Netstat DRop/-D command” on page 474.

Restriction: You can use this command only if the MVS.VARY.TCPIP.DROP security product resource profile is defined and the user ID associated with the DROP command is permitted for CONTROL access to this resource.

For dropping a single TCP connection or UDP socket:
You can use this command with the *connid* or CONNECTION parameters to terminate the specific TCP/IP socket endpoint that is identified by its connection number, *connid*. You can determine the connection number from the Conn column in the Netstat COnn/**-c** or Netstat TELnet/**-t** display. This version of the command is the console environment equivalent of the Netstat DRop/**-d** commands in the TSO and z/OS UNIX environments.

When the command is issued against a socket endpoint, any outstanding or subsequent socket calls that refer to the dropped socket terminate with a negative return code. The socket endpoint that you drop can be a listening TCP server socket endpoint, a fully connected TCP socket (either server or client connection endpoint), or a UDP socket endpoint. When you drop a TCP connection or UDP endpoint, the associated socket does not close. The application that owns the associated socket is responsible for closing the socket.

Tips:

- You can use this command when you do not want to stop the server itself, but only want to drop an individual TCP connection with that server.
- You can use this command to stop old TCP connections if they prevent a server from being restarted. This is sometimes necessary when the server does not enable the SO_REUSEADDR socket option before binding to its well-known port.

If you want to terminate all socket activities from a specific sockets application, terminate the application by using the appropriate mechanism that is provided by the application. This command can have unpredictable results if it is issued against a listening socket or UDP socket. Some applications might not handle the subsequent socket errors as expected. To drop all connections that are associated with a server application without terminating the listening socket, use the form of the VARY DROP command specifying a PORT or JOBNAME, as described in the following information.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──┬─DRop,─────┬──┬─connid──────────────┬──────►◄
                     └─procname─┘     └─CMD=DRop,─┘  └─CONNection=──connid─┘
```

Parameters:

procname
: The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* value is specified, the request will fail with an error message.

CMD=DRop or DRop
: Synonymous syntax for parameter used to drop a connection.

CONNection=*connid* **or** *connid*
: The *connid* value is a required parameter. It can be specified by itself or as the value of the CONNECTION parameter. Issue the Netstat COnn/**-c** command

or the DISPLAY TCPIP,,NETSTAT,CONN command to obtain the connection identifier for the TCP/IP socket connection that you want to drop.

For dropping all TCP connections associated with a TCP/IP server:
You can drop all TCP connections associated with a TCP/IP server using the VARY TCPIP,,DROP command by specifying filter parameters to identify the server whose TCP connections are to be terminated. If more than one server matches the filter criteria, you must specify additional parameters (for example, JOBNAME and ASID) to identify which TCP connections of the server will be dropped. When a TCP connection is dropped, the associated socket does not close. The application that owns the associated socket is responsible for closing the socket. The following message is issued to indicate that the command has completed processing:

```
EZD2013I numconn  CONNECTIONS WERE  SUCCESSFULLY DROPPED
```

Tip: Shut down the server before issuing this command to prevent new connections from using this server. For Sysplex Distributor connections, you can issue a VARY TCPIP,,SYSPLEX,QUIESCE command. For Load Balancing Advisor (LBA) connections, issue a MODIFY LBAGENT,QUIESCE command.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──┬─DRop,─────┬──────────────────────────────►
                     └─procname─┘     └─CMD=DRop,─┘

►──┬─POrt=portnum──┬─────────────────────────────────┬─┬─────────────────────►◄
   │               └─,JOBNAME=jobname──┬────────────┬─┘ │
   │                                   └─,ASID=asid─┘   │
   └─JOBNAME=jobname──┬────────────┬────────────────────┘
                      └─,ASID=asid─┘
```

Parameters:

procname
The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* value is specified, the request will fail with an error message.

CMD=DRop or DRop
Synonymous syntax for parameter that is used to drop a connection.

POrt=*portnum*
The port number parameter is an integer in the range 1- 65535.

Servers that are bound to this port number will have all their TCP connections reset. If the *portnum* value specifies a port that has more than one instance of a server bound to it with either a different *jobname* or *asid* value, then either the JOBNAME value, or the JOBNAME and ASID values must be specified to identify a unique server instance for which connections will be dropped.

JOBNAME=*jobname*
The *jobname* value specifies the MVS job name of the server with which the Drop command is to be associated.

- If the JOBNAME parameter is specified without the PORT keyword, then all servers with this *jobname* value will have their TCP connections dropped regardless of the port they are bound to.

- If the *jobname* value specifies a job name that has more than one instance of a server with that job name but that has a different *asid* value, then the ASID parameter must also be specified and all server instances that have a matching job name and address space ID will have their TCP connections dropped, regardless of the port they are using.
- The environment in which the server runs determines the job name that is to be associated with a particular server application.
- The *jobname* value can be up to 8 characters in length.

ASID=*asid*
The *asid* value specifies the hexadecimal address space ID associated with the server whose TCP connections are to be dropped. If more than one instance of that application is found and the *jobname* value is not unique, you must specify an *asid* value to drop TCP connections for all server instances that match this job name and *asid* value.

Examples:
The following examples are about dropping TCP/IP socket connections.

- The first example is directed to a TCP/IP address space started by the identifier TCPPROC and demonstrates how to drop a TCP connection number 5001:
```
VARY TCPIP,TCPPROC,CMD=DROP,CONNECTION=5001
```
- The next example assumes there is only one TCP/IP address space and demonstrates how to drop a UDP connection number 6001:
```
VARY TCPIP,,CMD=DROP,CONNECTION=6001
```
- This example indicates how to drop all the TCP connections associated with a server listening on port 75, with job name JOBSRVR1:
```
VARY TCPIP,,CMD=DROP,PORT=75,JOBNAME=JOBSRVR1
```
- This example indicates how to drop all the TCP connections associated with a server listening on port 75, with job name JOBSRVR1 in address space 15:
```
VARY TCPIP,,CMD=DROP,PORT=75,JOBNAME=JOBSRVR1,ASID=15
```
- This example indicates how to drop all the TCP connections associated with a server with job name JOBSRVR1 in address space 15, regardless of port:
```
VARY TCPIP,,CMD=DROP,JOBNAME=JOBSRVR1,ASID=15
```

VARY TCPIP,,OBEYFILE

Use the VARY TCPIP,,OBEYFILE command to update TCP/IP profile configuration statements to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TCP/IP address space.

Guidelines

Before activating new configuration statements with the VARY TCPIP,,OBEYFILE command, use the VARY TCPIP,,SYNTAXCHECK command to verify that the configuration statements specified by the **DSN=** or *datasetname* parameter are free of syntax errors. For information about the VARY TCPIP,,SYNTAXCHECK command, see “VARY TCPIP,,SYNTAXCHECK” on page 264.

See the z/OS Communications Server: IP Configuration Reference for information about how different parameter updates take effect with Obeyfile processing.

Format:

```
►►──Vary ──TCPIP──,──┬────────────┬──,──┬─Obeyfile,─────┬──┬─datasetname──────┬───►◄
                     └─procname───┘     └─CMD=Obeyfile,─┘  └─DSN=──datasetname─┘
```

Parameters:

procname
The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* value is specified, the request will fail with an error message.

CMD=OBEYFILE or OBEYFILE
Specify this parameter to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TCP/IP address space. These changes are in effect until the TCP/IP cataloged procedure is started again or until another VARY OBEYFILE overrides them. Put your changes in the data set specified by the *datasetname* value. You can maintain different data sets that contain a subset of the TCP/IP configuration statements and activate them while TCP/IP is running.

DSN=*datasetname* **or** *datasetname*
The *datasetname* value is required after specifying the OBEYFILE parameter. The *datasetname* value is the name of a data set that contains TCP/IP profile configuration statements. The *datasetname* value must be a cataloged and fully-qualified data set name that is specified without any quotation marks. The *datasetname* value can be either a sequential data set or a member in a PDS. The *datasetname* value cannot be a z/OS UNIX file. The **DSN=** parameter cannot be a z/OS UNIX file or a TCPIP.DATA data set. For more information about updating TCPIP.DATA configuration statements, see the information about dynamically changing TCPIP.DATA statements in z/OS Communications Server: IP Configuration Reference.

Examples:
The following examples are about updating system operation and network configuration information without stopping and restarting the TCP/IP address space.

- The first example is directed to a TCP/IP address space started by the identifier TCPPROC, and assumes the sequential data set USER99.TCPIP.OBEYFIL1 contains TCP/IP profile configuration statements:

  ```
  VARY TCPIP,TCPPROC,CMD=OBEYFILE,DSN=USER99.TCPIP.OBEYFIL1
  ```

- The next example assumes there is only one TCP/IP address space and that OBEYFIL2 is a member of the PDS USER99.TCPIP and contains TCP/IP profile configuration statements:

  ```
  VARY TCPIP,,O,USER99.TCPIP(OBEYFIL2)
  ```

Usage:
Tip

Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.OBEYFILE.

VARY TCPIP,,OSAENTA

Use the VARY TCPIP,,OSAENTA command to control the OSA-Express network traffic analyzer (OSAENTA) tracing facility in the OSA-Express adapter. You can use this command to select frames as candidates for tracing and for subsequent analysis. OSAENTA traces are recorded externally using the TRACE command. See the z/OS Communications Server: IP Diagnosis Guide for information about the steps required to perform an OSAENTA trace.

The OSAENTA command consists of two parts.

- The first part defines the OSA that is to be traced and the characteristics of the tracing.
- The second part turns tracing on or off, or clears the trace settings.

The tracing characteristics are identified by filters that specify under which conditions a frame should be traced. A frame must meet all of the conditions specified on the OSAENTA commands for it to be traced. For example, if the OSAENTA command identifies PROTOcol=TCP and PORTNum=21, then only IP packets that have both a protocol of TCP and a port number of 21 are traced. Only one value can be specified for a given filter each time the OSAENTA command is issued.

Multiple OSAENTA commands can be included in a profile data set and can control tracing for multiple OSAs. The filters on multiple OSAENTA commands are cumulative for a given OSA. As each OSAENTA command is issued with filters, those filters are added to the filters that are already in effect for that OSA. By using multiple OSAENTA commands, multiple filter values can be assigned to each filter. There is a limit of eight filter values for each filter for each OSA. For example, you can specify up to eight IP protocols, up to eight VLAN IDs, and so on. For IP addresses, you can specify up to eight IPv4 addresses and up to eight IPv6 addresses. If a frame matches any of the values for that filter, it meets the condition of that particular filter. For example, if you specify IPaddr=9.67.1.1, PROTOcol=TCP, and PORTNum=21 on one OSAENTA command for OSA1, and you specify IPaddr=9.67.1.2 on another OSAENTA command for OSA1, then all frames sent to either IP address 9.67.1.1 or 9.67.1.2 with a protocol of TCP and a port number of 21 are traced.

The OSAENTA command dynamically defines a QDIO interface to the OSA-Express adapter being traced, called an OSAENTA interface. That interface is used exclusively for capturing OSA-Express network traffic analyzer traces.

Security Rule: The OSAENTA command enables an installation to trace data from other hosts connected to an OSA. The trace data collected should be considered confidential and TCPIP system dumps and external trace files that contain this trace data should be protected. The OSAENTA command is protected by the operating system security product. The name of the protected OPERCMDS resource is MVS.VARY.TCPIP.OSAENTA.

Tips:

- You can specify the parameters for this statement in any order.
- If a keyword on a given command is specified multiple times, the last value specified is used.
- If an error is found while parsing the OSAENTA command, an error message is generated and the command is ignored.

Format:

```
►►—Vary —TCPIP—,——————————————,—OSAENTA————————————————————————————►◄
                └─procname─┘           └─┤ Command ├─┘
```

Command:

```
                                         ┌──────────────────────────┐ (1) (2)
├──,──PORTNAME=osa_port_name──,──┬─ON──┬─┴─┬─┤ Trace Parameters ├──┬─┴────────────┤
                                 ├─OFF─┤   ├─┤ Protocol Type ├─────┤
                                 └─DEL─┘   ├─┤ IP Address ├────────┤
                                           ├─┤ Packet Port ├───────┤
                                           ├─┤ Device Identifier ├─┤
                                           ├─┤ Ethernet Type ├─────┤
                                           ├─┤ MAC Address ├───────┤
                                           └─┤ VLAN ID ├───────────┘
```

Trace Parameters:

```
   ┌─,FULL──────────────────────────────┐
├──┼────────────────────────────────────┼──┬──────────────┬──────────────────────►
   │                ┌─224─────────────┐ │  └─,CLEARfilter─┘
   └─,ABBREV──=─────┴─abbrev_length───┴─┘

                                     ┌─,DISCARD=EXCEPTION──────┐
►──┬─────────────────────────────┬───┼─────────────────────────┼─────────────────►
   │          ┌─1024──────────┐  │   ├─,DISCARD=ALL────────────┤
   └─,DATA=───┴─trace_amount──┴──┘   ├─,DISCARD=NONE───────────┤
                                     └─,DISCARD=discard_code───┘

                                         ┌─,NOFILTER=NONE─┐
►──┬─────────────────────────────────┬───┼────────────────┼──────────────────────►
   │            ┌─2147483647──────┐  │   └─,NOFILTER=ALL──┘
   └─,FRAMES=───┼─────────────────┼──┘
                └─trace_count─────┘

►──┬──────────────────────────────┬──────────────────────────────────────────────┤
   │          ┌─10080──────────┐  │
   └─,TIME=───┼────────────────┼──┘
              └─trace_time─────┘
```

Protocol Type:

```
   ┌─,PROTOcol──=──*────────────────────┐
├──┼────────────────────────────────────┼────────────────────────────────────────┤
   ├─,PROTOcol──=──TCP──────────────────┤
   ├─,PROTOcol──=──UDP──────────────────┤
   ├─,PROTOcol──=──ICMP─────────────────┤
   ├─,PROTOcol──=──ICMPV6───────────────┤
   └─,PROTOcol──=──protocol_number──────┘
```

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report

IP Address:

```
   ┌─,IPaddr──=──*──────────────────────────────┐
├──┼──────────────────────────────────────────────┼─────────────────────────┤
   │                                 ┌─/32──────────┐  │
   ├─,IPaddr──=──ipv4_address────┼──────────────┼──┤
   │                                 └─/num_mask_bits─┘  │
   │                                 ┌─/128─────────┐  │
   └─,IPaddr──=──ipv6_address────┼──────────────┼──┘
                                     └─/prefix_length─┘
```

Packet Port:

```
   ┌─,PORTNum──=──*──────────┐
├──┼──────────────────────────┼───────────────────────────────────┤
   └─,PORTNum──=──port_number─┘
```

Device Identifier:

```
   ┌─,DEVICEID──=──*────────┐
├──┼─────────────────────────┼────────────────────────────────────┤
   └─,DEVICEID──=──device_id─┘
```

Ethernet Type:

```
   ┌─,ETHType──=──*────────────┐
├──┼────────────────────────────┼─────────────────────────────────┤
   ├─,ETHType──=──IPV4──────────┤
   ├─,ETHType──=──IPV6──────────┤
   ├─,ETHType──=──ARP───────────┤
   ├─,ETHType──=──SNA───────────┤
   └─,ETHType──=──ethernet_type─┘
```

MAC Address:

```
   ┌─,MAC──=──*────────────┐
├──┼────────────────────────┼─────────────────────────────────────┤
   └─,MAC──=──mac_address─┘
```

VLAN ID:

```
   ┌─,VLANID──=──*───────┐
├──┼──────────────────────┼───────────────────────────────────────┤
   ├─,VLANID──=──vlan_id─┤
   └─,VLANID──=──ALL─────┘
```

Notes:

1 Each option can be specified only once. The order of options is not important.

2 You must also issue the MVS TRACE command for component SYSTCPOT to activate the OSAENTA trace. See the z/OS Communications Server: IP Diagnosis Guide for details.

Parameters:

procname
The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* value is specified, the request fails with an error message.

OSAENTA
Specifies that this command is for OSAENTA information.

PORTNAME=*osa_port_name*
Specifies the name of the OSA port for which tracing is needed. This is the same port name that is defined on the VTAM TRLE statement PORTNAME keyword. This parameter is required. Specifies the name of the OSA port for which tracing is required. This is the same port name that is either defined on the VTAM TRLE statement PORTNAME keyword or is dynamically created by VTAM for OSX interfaces (configured with the CHPID parameter) or for OSM interfaces. This parameter is required. For more information about OSM and OSX interfaces, see TCP/IP in an intra ensemble network in z/OS Communications Server: IP Configuration Guide.

Tip:

- You are not required to also define OSA-Express to TCP/IP using the DEVICE/LINK or INTERFACE statement or activate it on the tracing stack in order to collect trace data from other stacks using that OSA-Express. For an OSX interface configured with the CHPID parameter or for an OSM interface, specify the port name according to the VTAM naming convention for these dynamic TRLEs, and VTAM will dynamically create the TRLE when you activate the OSAENTA interface. For details about the naming convention for these dynamically generated TRLEs, see Defining an OSA-Express device to z/OS Communications Server using QDIO in z/OS Communications Server: SNA Network Implementation Guide.

Restriction: Multiple stacks cannot use the tracing function concurrently for a given OSA.

FULL
Specifies that the entire frame is to be traced, if possible. (An OSA might limit the amount of data that is actually traced.)

ABBREV={*abbrev_length***|224}**
Specifies the amount of data that is to be traced for each frame.

- You can specify a data length in the range 64–65472 or use the default value 224. The value is rounded up to the next 32 byte boundary.
- The ABBREV parameter can be used to control the volume of data stored in the trace buffers and file.
- The actual amount of data traced might be limited by the OSA.

Guideline: Use a large value or the FULL parameter if you want to maximize the amount of data traced for each packet because TCP segmentation offload packets are traced before the packet is segmented and can be larger than the largest frame size on the LAN. See TCP segmentation offload in the z/OS Communications Server: IP Configuration Guide for information about which parameters affect the size of TCP segmentation offload packets.

Restriction: Many OSA models limit the amount of data recorded to 120 bytes. The OSAENTA command accepts a larger size (or FULL) without any errors, but the actual trace entries are as if ABBREV=120 is specified.

CLEARFILTER

Clears any previous OSAENTA trace filters for the port specified by the *osa_port_name* value.

Guideline: If you specify the CLEARFILTER parameter and the OSAENTA interface is active, either all are frames traced or no frames are traced, depending on the setting of the NOFILTER parameter.

Tip: The CLEARFILTER parameter clears all filters. To clear all values for a single filter, use the OSAENTA command and specify an asterisk (*) for the filter that you want to use.

DATA={*trace_amount*|1024}

Specifies the number of megabytes (MB) of data to be collected before stopping the trace.

- The minimum value is 1 MB
- The default value is 1024 MB
- The maximum value is 2 147 483 647 MB

If a value of 0 is specified, then the maximum value is set.

Result: If the OSAENTA interface is inactive, then the limit specified by the DATA parameter takes effect when the OSAENTA trace is enabled with the ON parameter. If the OSAENTA interface is active and the DATA parameter value is modified, then the stack resets the data counter to 0 and puts the new DATA limit into effect.

DEL

Removes the OSAENTA interface definition. The OSAENTA interface must be inactive for you to specify the DEL parameter. To dactivate the OSAENTA interface, you can respecify the OSAENTA statement with the OFF parameter, or use the VARY TCPIP,,OSAENTA command with the OFF parameter.

DEVICEID={*device_id*|*}

Specifies the 8-digit hexadecimal value that identifies a host that is sharing the OSA. This value is in the form *csmfclua* where the digits have the following values:

- *cs* – The channel subsystem ID for this datapath device.
- *mf* – The LPAR multiple image facility ID for the LPAR using this datapath device.
- *cl* – The control unit logical identifier for this datapath device.
- *ua* – The unit address for this datapath device.

Each identifier is a 2-digit hexadecimal value in the range 00–FF.

If the frame was either inbound or outbound to the host that is identified by the *device_id* value, then the frame meets the criteria for this filter. If the DEVICEID option has been omitted or if an asterisk (*) is specified, then all packets meet the criteria for this filter.

Tip: You can obtain the *device_id*values for any user of the OSA by using the hardware management console (HMC). For a datapath device that is active on a z/OS stack, you can obtain the *device_id* value for that datapath device from message IST2190I of the output from the DISPLAY NET,TRL,TRLE= command.

DISCARD={ALL|EXCEPTION|NONE|*discard_code*}

Specifies which frames that were discarded by the OSA-Express device should be traced. Discarded frames include frames that the OSA-Express device could not transmit outbound or could not forward inbound. Discarded frames that match the DISCARD= setting are traced whether they match any filters that are in effect or not.

ALL

All frames discarded by the OSA-Express device are traced. This includes both exception conditions and more expected discards, such as ARP packets received for non-registered IP addresses or packets for non-supported Ethernet types.

EXCEPTION

Frames discarded by the OSA-Express device for exception conditions are traced. These are frames that are typically discarded for anomalous conditions. See the following examples of anomalous conditions:

- An inbound IP packet destined for an IP address that is not registered with the OSA-Express device and no PRIROUTER or SECROUTER parameter is in effect.
- An outbound IP packet that could not be delivered because no storage was available within the OSA-Express device.

NONE

No discarded frames are traced.

discard_code

Frames discarded for the reason specified by the *discard_code* value are traced. Use this option only under the direction of IBM Service personnel. Values in the range 1-4087 are accepted. Up to eight discard codes can be active for one OSA-Express device.

Rule: As with filters, the DISCARD keyword can be specified on multiple OSAENTA statements. The ALL and NONE options reset any previous DISCARD values that are in effect; the EXCEPTION option or a discard code resets a current setting of ALL or NONE. EXCEPTION and *discard_code* options are cumulative for a given OSA. If EXCEPTION and *discard_code* options are specified on multiple OSAENTA statements, all frames discarded for exception conditions and all frames discarded for any of the discard codes that are in effect are traced. When the EXCEPTION option is in effect, a limit of seven discard codes can be active for one OSA-Express device.

Result: A frame can be traced twice; once when the packet is passed to the OSA-Express device, and again as a dropped packet during the processing of the packet.

Guideline: To reset the current set of active discard codes, specify the value DISCARD=ALL or DISCARD=NONE followed by OSAENTA statements with the required DISCARD options that you want to specify.

ETHType={IPV4|IPV6|ARP|SNA|*ethernet_type*|*}

Specifies the Ethernet frame type to be traced. This can be specified as one of the literals IPV4, IPV6, ARP, SNA, or as a hexadecimal number in the range 0600–FFFF (IPV4=0800, IPV6=86DD, ARP=0806, and SNA=80D5). If the ETHType parameter has been omitted or if an asterisk (*) is specified, then all packets meet the criteria for this filter.

FRAMES={*trace_count*|2147483647}

Specifies the number of frames to be recorded before tracing is stopped. The minimum value is 100 frames. The maximum value is 2147483647 frames. If the value 0 is specified, then the maximum value is set.

Result: If the OSAENTA interface is inactive, then the FRAMES parameter limit takes effect when the OSAENTA trace is enabled with the ON parameter. If the OSAENTA interface is active and the FRAMES parameter value is modified, then the stack resets the frame counter to 0 and puts the new FRAMES parameter limit into effect.

IPaddr={*ipv4_address***[/***num_mask_bits***]|***ipv6_address***[/***prefix_length***]|*}**

Specifies an IP address (either a 32-bit IPv4 address in dotted decimal notation, or a 128-bit IPv6 address colon hexadecimal notation) to be compared with both the source and destination addresses of inbound and outbound packets. If either the source or the destination address of a packet matches the specified IP address, the frame meets the criteria for this filter. If the IPaddr option is omitted or if an asterisk (*) is specified, then all packets meet the criteria for this filter. If the IPaddr filter is specified, then only frames that contain IP packets or ARP packets are subject to tracing.

If an IPv4 address is specified, then you can specify a /*num_mask_bits* value in the range 1–32 to designate a subnet. The default number of bits is 32.

If an IPv6 address is specified, then you can specify an optional *prefix_length* value in the range 1–128. The default *prefix_length* value is 128.

MAC={*mac_address***|*}**

Specifies the 12 hexadecimal digits of the MAC address. The address is compared with both the source and destination MAC addresses of both inbound and outbound frames. If either the source or destination address of a frame matches the specified MAC address, the frame meets the criteria for this filter. If the MAC option has been omitted or if an asterisk (*) is specified, then all packets meet the criteria for this filter.

NOFILTER=ALL|NONE

Specifies the filtering behavior when all filters (DEVICEID, MAC, ETHTYPE, VLANID, IPADDR, PROTOCOL and PORTNUM) have been cleared or are inactive. This condition can exist if no filters have been specified, if CLEARFILTER is specified, or when the current setting for every filter is set to an asterisk (*). When the NOFILTER=ALL setting is in effect, all packets are traced. When the NOFILTER=NONE setting is specified, no packets are traced. The NOFILTER parameter applies only to packets that were not discarded by the OSA-Express device. The DISCARD parameter controls tracing of discarded packets.

Guideline: If you clear filters using the CLEARFILTER parameter with the OSAENTA interface active, and specify NOFILTER=ALL, ensure that you also specify sufficient new filters. The trace buffers are likely to fill up quickly if you clear all filters without setting new filters to filter out an adequate percentage of the packets.

OFF

Disables OSA tracing for the port specified by the *osa_port_name* value by stopping the OSAENTA interface. The trace parameters and filters remain in effect if the OSAENTA trace is subsequently re-enabled.

ON Enables OSA tracing for the port specified by the *osa_port_name* value by starting the OSAENTA interface using the OSAENTA trace parameters and filters that are currently in effect. If the OSAENTA interface is already active, then the ON keyword causes the stack to reset the active counters on the DATA, FRAMES, and TIME parameter limits.

Guideline: Ensure that you have specified sufficient trace filters before starting the trace. The trace buffers are likely to fill very quickly if you activate the trace with no filters or with a set of filters that does not filter a significant percentage of the packets.

PORTNum={*port_number* |*}
Specifies a port number in the range 1–65535. The port number is compared with the destination or source port of both inbound and outbound packets. If the port of a packet is the same as the specified port number, then the frame meets the criteria for this filter. This comparison is performed only for packets using either the TCP or UDP protocol; frames using other protocols are not traced when a port filter is in effect. If the PORTNum parameter is omitted or if an asterisk (*) has been specified, then all packets meet the criteria for this filter. If the port filter is used, only frames containing IP packets are subject to tracing.

IPSec Encapsulating Security Payload (ESP) packets cannot be traced by specifying a port number because the TCP or UDP headers are encrypted.

PROTOcol={TCP|UDP|ICMP|ICMPV6|*protocol_number*|*}
Specifies the IP protocol type to be traced. This can be specified as one of the literals TCP, UDP, ICMP, ICMPV6, or as a number in the range 0–255 (ICMP=1, TCP=6, UDP=17, ICMPV6=58). If the PROTOcol parameter is omitted or if an asterisk (*) has been specified, then all packets meet the criteria for this filter. If a PROTOcol value is specified and the frame does not contain an IP protocol packet, then the frame is not traced. If the PROTOcol filter is used, only frames containing IP packets are subject to tracing.

Rule: For encapsulated packets, OSAENTA collects packets based on whether the specified protocol filter matches the outermost packet protocol. For example, if you specify TCP as the protocol filter and a TCP packet is received that is encapsulated in an IPSec packet with protocol 50, this TCP packet is not collected. You must specify Protocol 50 to collect these packets.

TIME={*trace_time*|10080}
Specifies the number of minutes that trace records are recorded before stopping. The minimum value is 1 minute. The maximum value is 10 080 minutes (7 days). If a value 0 is specified, then the maximum value is set.

Result: If the OSAENTA interface is inactive, then the TIME parameter limit takes effect when the OSAENTA trace is enabled with the ON parameter. If the OSAENTA interface is active and the TIME parameter value is modified, then the stack resets the time counter to 0 and puts the new TIME parameter limit into effect.

VLANID={*vlan_id*|*|ALL}
Specifies a VLAN identifier value, which is a decimal number in the range 0–4094. The ALL keyword specifies that all frames that have a VLAN tag are included. If the VLANID parameter has been omitted or if an asterisk (*) is specified, then all frames meet the filter criteria. If a VLAN identifier is specified and the frame does not contain a VLAN tag or does not match the VLAN identifier, then the frame is not traced.

The OSAENTA statements are cumulative for a given OSA-Express adapter, and any subsequent OSAENTA statement processed adds to the filters that are already in effect for that OSA. To actually change a value for a given filter, several options are available:

- Define an OSAENTA statement with a filter value specified by an asterisk (*), effectively deleting all values for that one filter entirely. Then define subsequent OSAENTA statements with the new filter values.
- Define an OSAENTA statement with the CLEARFILTER parameter, which removes all existing filters, and subsequently specify the entire list of filter attributes that you want to use.

Tip: If the trace is currently enabled, the trace continues to run while each filter is modified or added. This can become an issue when changing a value for a given filter as previously described. Because both options involve deleting current filters, more data than you want is being traced during this time. For a more efficient trace, first disable the trace (define an OSAENTA statement with the OFF parameter) before changing filter values.

Examples:
To trace all the packets for a particular application port, enter the following OSAENTA command:

VARY TCPIP,,OSAENTA,PORTNAME=osa4,ON,PORTNUM=21

Usage:

- You can use the Netstat DEvlinks/-d command to display the current OSAENTA trace settings.
- When the DATA, FRAMES, or TIME values are exceeded, the stack disables the OSAENTA trace, but this does not happen immediately. Trace records from the OSA continue to be recorded until the stack has successfully contacted the adapter to stop the OSAENTA trace.
- To verify that the Ctrace component SYSTCPOT is active for a stack, issue DISPLAY TRACE,COMP=SYSTCPOT,SUB=(*tcpip_procname*)
- To write the data to the external writer, use the MVS TRACE,CT,WTRSTART=*writer_procedure* command to start the writer and the TRACE CT,ON,COMP=SYSTCPOT,SUB=(*tcpip_procname*) command to connect to the writer.
- The last buffer trace data are not written to the external writer until the writer has been disconnected from TCPIP and stopped.
- The TRACE CT,OFF,COMP=SYSTCPOT,SUB=(*tcpip_procname*) command stops the recording of trace data into TCPIP buffers and to the external writer. It does not stop the receipt of trace data from the OSA. A TRACE ON command is required to start recording of the trace data into the buffers. To halt the receipt of trace data from the OSA, specify the OSAENTA statement with the OFF parameter, or use the VARY TCPIP,,OSAENTA command with the OFF parameter.
- Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.OSAENTA.

The following differences exist between OSAENTA and PKTtrace:

- The PKTTRACE command can collect only data for a single TCPIP stack. The OSAENTA command can collect data for other stacks sharing the OSA.
- The PKTtrace data collection starts immediately. The OSAENTA data collection is not started until the ON parameter is used.
- Each PKTtrace command or statement is one set of filters. OSAENTA command filters accumulate across multiple OSAENTA commands or statements.

VARY TCPIP,,PKTTRACE

Use the VARY TCPIP,,PKTTRACE command to set up tracing.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──PKTtrace──┬─────────────┬──────────►◄
                     └─procname─┘               └─┤ Command ├─┘
```

Command:

```
                                                   ┌────────────────────────────┐ (1) (2)
├──,──┬──────────────────────────┬──┬─ON────┬────▼─┬─┤ Packet Length ├──────────┬┴────────────┤
      ├─LINKName──=──*──,────────┤  ├─OFF───┤      ├─┤ Protocol Type ├──────────┤
      ├─LINKName──=──link_name──,┤  └─CLEAR─┘      ├─┤ Packet Dest Address ├────┤
      ├─INTFName──=──*──,────────┤                 ├─┤ Packet Source Port ├─────┤
      └─INTFName──=──intf_name──,┘                 ├─┤ Packet Dest Port ├───────┤
                                                   ├─┤ Packet Port Number ├─────┤
                                                   └─┤ Packet Discard Code ├────┘
```

Packet Length:

```
    ┌─,FULL──────────────────────────────┐
├───┼────────────────────────────────────┼─────────────────────────────────────┤
    │               ┌─=──200──────────┐  │
    └─,ABBREV───────┼─────────────────┼──┘
                    └─=──abbrev_length┘
```

Protocol Type:

```
    ┌─,PROT──=──*─────────────────┐
├───┼─────────────────────────────┼────────────────────────────────────────────┤
    ├─,PROT──=──TCP───────────────┤
    ├─,PROT──=──UDP───────────────┤
    ├─,PROT──=──ICMP──────────────┤
    ├─,PROT──=──ICMPV6────────────┤
    └─,PROT──=──protocol_number───┘
```

Packet Dest Address:

```
    ┌─,IPaddr──=──*──────────────────────────────────────────────┐
├───┼────────────────────────────────────────────────────────────┼─────────────┤
    │                           ┌─,SUBNet──=──255.255.255.255─┐  │
    ├─,IPaddr──=──ipv4_address──┼─────────────────────────────┼──┤
    │                           ├─,SUBNet──=──subnet_mask─────┤  │
    │                           └─/num_mask_bits──────────────┘  │
    │                           ┌─/128──────────────┐            │
    └─,IPaddr──=──ipv6_address──┼───────────────────┼────────────┘
                                └─/prefixLength─────┘
```

Packet Source Port:

```
   ┌─,SRCPort──=──*───────────┐
├──┤                          ├──────────────────────────────────────────┤
   └─,SRCPort──=──source_port─┘
```

Packet Dest Port:

```
   ┌─,DESTport──=──*────────────────────┐
├──┤                                    ├────────────────────────────────┤
   └─,DESTport──=──destination_port─────┘
```

Packet Port Number:

```
   ┌─,PORTNUM──=──*───────────┐
├──┤                          ├──────────────────────────────────────────┤
   └─,PORTNUM──=──port_number─┘
```

Packet Discard Code:

```
   ┌─DISCard=NONE────────┐
├──┤                     ├───────────────────────────────────────────────┤
   ├─DISCard=*───────────┤
   ├─DISCard=ALL─────────┤
   └─DISCard=reason_code─┘
```

Notes:

1 Each option can be specified only once. The order of options is not important.

2 The MVS TRACE command must also be issued for component SYSTCPDA to activate the packet trace. See the z/OS Communications Server: IP Diagnosis Guide for details.

Parameters:

procname
: The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* value is specified, the request will fail with an error message.

PKTtrace
: Specifies this command is for PKTTRACE information.

LINKName=*link_name*
INTFName=*intf_name*

Specifies the name of the network interface that is defined on a preceding LINK or INTERFACE statement. If the LINKName/INTFName parameter is omitted or if an asterisk (*) is specified for either parameter, the PKTTRACE parameters apply to all IPv4 and IPv6 interfaces.

To facilitate defining packet tracing when many interfaces are involved, use the PKTTRACE statement with the LINKName=* or INTFName=* option to define packet tracing characteristics for the majority of the interfaces. Then use individual PKTTRACE statements with specific LINKName/INTFName parameters for each interface that must be defined differently from the majority.

The LINKName and INTFName parameters are interchangeable.

ON Turns on packet tracing, clears all settings previously defined and refreshes just the default settings.

If you use LINKName=* or INTFName=* and all other parameters are defaults, even if the defaults are specified, the command results replaces any existing trace structures for all existing IPv4 and IPv6 interfaces.

If you use LINKName=*link_name* or INTFName=* and another non-default parameter, the command results are added to any existing trace structures. However, if the existing trace structure for *link_name* or *intf_name* is all defaults, the existing trace structure will be discarded.

OFF

Disables packet tracing for the interfaces specified and removes the characteristics defining how they should be traced.

If LINKName=* or INTFName=* and all other parameters are defaults, all trace structures are deactivated and removed from all existing IPv4 and IPv6 interfaces.

If LINKName=* or INTFName=* and PROT=UDP, all trace structures for all resources are analyzed; any matches are removed. If no trace structures remain, trace is deactivated for that resource.

If LINKName=*link_name* or INTFName=*intf_name* and there are no other parameters, all trace structures for *link_name* or *intf_name* are deactivated and removed.

If LINKName=*link_name* and IP=127.0.0.1 or INTFName=*intf_name* and IP=::1, that particular trace structure is removed if it is found. If there is only one trace structure, then that structure is removed and trace is deactivated for that resource.

CLEAR

Disables packet tracing for the interfaces specified and removes the characteristics that define how the interfaces should be traced.

FULL

Specifies that the entire IP packet is to be traced.

ABBREV

Specifies that a truncated portion of the IP packet is to be traced. You can specify a length in the range 0 - 65,535 or use the default of 200. The ABBREV parameter can be used to reduce the volume of data stored in the trace file.

Note: The protocol headers are always included even if they exceeds the ABBREV value.

PORTNUM

Specifies a port number that is compared with the destination port and source port of inbound and outbound packets. You can use this parameter instead of using the SRCPORT and DESTPORT parameters. The port number is an integer in the range 1 - 65,535. If the destination port or source port of a packet is the same as the specified port number, the packet is traced. This comparison is performed only for packets that use the TCP or UDP protocol; packets using other protocols are not traced. If the PORTNUM parameter is omitted and the

SRCPORT and DESTPORT parameters are also omitted, the port numbers of packets are not checked. If an asterisk (*) is specified, packets of any protocol and of any destination or source port are traced.

IPSec Encapsulating Security Payload (ESP) packets cannot be traced by port number because the TCP or UDP headers are encrypted.

Guideline: SRCPORT and DESTPORT parameters should not be specified on the same PKTTRACE statement as the PORTNUM parameter. When the PORTNUM parameter is specified after the DESTPORT or SRCPORT parameters, the DESTPORT and SRCPORT parameters are ignored.

PROT

Specifies the protocol type to be traced. This can be specified as one of the literals TCP, UDP, ICMP, or ICMPV6, or as a number in the range 1 - 255 (ICMP=1, TCP=6, UDP=17, and RAW=255). If the PROT parameter is omitted or an asterisk (*) is specified, packets of any protocol are traced.

IPaddr

Specifies an IP address (either a 32-bit IPv4 address in dotted decimal notation, or a 128-bit IPv6 address colon hexadecimal notation) that is compared with both the source and destination addresses of inbound and outbound packets. If either the source or destination address of a packet matches the specified IP address, the packet is traced. If the IP option is omitted, or an asterisk (*) is specified, then all IP addresses are traced.

If an IPv6 address is specified, then an optional *prefixLength* (range 1 - 128) is allowed. IPv4 addresses and IPv4-mapped IPv6 addresses are treated as equivalent addresses. The default *prefixLength* is 128. If an IPv4 address is specified, then /*num_mask_bits* can be used. The *num_mask_bits* and SUBNET values are mutually exclusive. An error message is displayed if both are coded.

SUBNET

Valid only with `IP=`*`ipv4_address`*. Specifies a subnet mask that applies to the host and network portions of the IP address specified on the `IP=`*`ipv4_address`* parameter. The subnet mask must be specified in dotted decimal notation and must be specified in conjunction with the `IP=`*`ipv4_address`* parameter. With an IPv4 address specified, the /*num_mask_bits* can be used. The *num_mask_bits* and SUBNET are mutually exclusive. An error message is displayed if both are coded.

SRCPORT

Specifies a port number that will be compared with the source port of inbound and outbound packets. The port number is an integer in the range 1 - 65,535. If the source port of a packet is the same as the specified port number, the packet is traced. This comparison is performed only for packets using either the TCP or UDP protocol; packets using other protocols are not traced. If the SRCPORT parameter is omitted, there is no checking of the source port of packets. If an asterisk (*) is specified, packets of any protocol and any source port are traced. If the SRCPORT and PORTNUM parameters are omitted, or if an asterisk (*) is specified for the SRCPORT parameter, the source port of packets is not checked.

IPSec Encapsulating Security Payload (ESP) packets cannot be traced by port number because the TCP or UDP headers are encrypted.

DESTPORT

Specifies a port number that will be compared with the destination port of inbound and outbound packets. The port number is an integer in the range 1 - 65,535. If the destination port of a packet is the same as the specified port number, the packet is traced. This comparison is performed only for packets tat use the TCP or UDP protocol; packets using other protocols are not traced. If the DESTPORT and PORTNUM parameters are omitted or if an asterisk (*) is specified for the DESTPORT parameter, the destination port of packets is not checked.

IPsec Encapsulating Security Payload (ESP) packets cannot be traced by port number because the TCP or UDP headers are encrypted.

DISCARD

Specifies the IP packet discard reason code of the packets which should be traced. All IP packets contain a discard reason code that is normally set to 0. When the TCP/IP stack decides to discard a packet, a specific discard reason code is set in this field. See IP Discard reason codes information in z/OS Communications Server: IP and SNA Codes for a list of all the discard reason codes. Normally, the TCP/IP stack does not trace discarded packets. You must specify a DISCARD value other than NONE in order to trace discarded packets.

NONE

Specifies that only IP packets that were not discarded should be traced. This is the default value.

***** The DISCARD parameter is not applied to the selection of packets. All packets are traced.

ALL

Specifies that IP packets with a nonzero discard reason code should be traced. Specifying this value results in tracing all discarded packets.

reason_code

Specifies that only IP packets with the specified discard reason code should be traced. Valid *reason_code* values are numbers in the range 4096 -20,479. The value 0 can also be specified, which is the equivalent of specifying DISCARD=NONE.

Tips:

- Specifying the SRCPORT, DESTPORT, IPADDR, PORTNUM, or PROTOCOL parameters might prevent malformed packets from being traced.
- A packet might be traced twice, once at the lower level IP layer when a packet arrives, and again as a discarded packet in an upper level protocol layer of TCP/IP.

You can use one packet trace profile statement per discard reason code. You can also specify a packet trace statement with the DISCARD=ALL option to trace all packets that are dropped. The other specified parameters further identify which discarded packets are traced. The following example collects packets with the discard reason code 4138 on all TCP and UDP packets that specify the PORT number 20.

```
PKTTRACE ON,DISCARD=4138,PORTNUM=20
```

Examples:
To trace all packets for a particular application port, enter the following two PKTTRACE commands:

```
v tcpip,,pkt,on,dest=21
v tcpip,,pkt,on,srcp=21
```

The two commands will capture all the packets received and all the packets sent for a particular port. If other options are specified, then they should be the same on both commands.

Usage:

- The results are cumulative when you issue multiple PKTTRACE commands. Use the NETSTAT DEvlinks (**netstat -d**) command to display the results. An IP packet is traced according to the first setting that matches. You might need to issue the CLEAR command to reset active PKTTRACE filters if existing filters are not needed before you enable new PKTTRACE filters
- Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.PKTTRACE.

VARY TCPIP,,PURGECACHE

Use the VARY TCPIP,,PURGECACHE command to delete the ARP cache entries or neighbor cache entries for an interface.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──PURGECache,name────────────────────►◄
                     └─procname─┘
```

Parameters:

name
: The interface name of the cache that is to be purged.

  If the *name* value matches an IPv4 interface name, the local ARP cache or the outboard OSA cache entries (for QDIO token ring and QDIO Ethernet) for that interface is purged. If the *name* matches an IPv6 interface name, the IPv6 neighbor cache for that interface is purged.

  Note:

  1. Purging of the OSA outboard cache entries requires a level of microcode that supports the Flush ARP table ARP Assist Option Request. When this command is issued against an IPv4 QDIO token ring or Ethernet interface and the OSA-Express device is shared by multiple stacks, then this command purges the ARP cache for all stacks that share the OSA (because an OSA-Express device maintains a single ARP cache for all stacks that share it).
  2. Translate entries are not deleted for ATM or LCS interfaces.

     For ATM, the following conditions apply:

     - PVC and ATMARP server entries are not deleted.
     - ACTIVE SVC entries are not deleted because TCP/IP processing periodically validates these entries.
     - A clear might be needed for SVC entries that are not ACTIVE. When the asynchronous clear completes, the entries are deleted.

Examples:

The following example shows how to use PURGECache.

- From TSO:

```
 netstat arp all
MVS TCP/IP NETSTAT CS V1R9 TCPIP Name: TCPCS
 Querying ARP cache for address 9.67.113.1
```

```
Interface: TR1 IBMTR: 000BC6AA1B88
Route info: 0000

Querying ARP cache for address 9.67.113.61
Interface: TR1 IBMTR: 08005A8B2EC7
Route info: 02A0
READY
```

- On MVS console:

```
v tcpip,,purgec,tr1
PROCESSING COMMAND: VARY TCPIP,,PURGEC,TR1
COMMAND PURGECACHE COMPLETED SUCCESSFULLY
PURGECACHE PROCESSED FOR LINK TR1
```

- From TSO:

```
  netstat arp all
MVS TCP/IP NETSTAT CS V1R9 TCPIP Name: TCPCS
Querying ARP cache for address 9.67.113.61
Interface: TR1 IBMTR: 08005A8B2EC7
Route info: 02A0
READY
```

Usage:
Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.PURGECACHE.

VARY TCPIP,,START or VARY TCPIP,,STOP

Use the VARY TCPIP,,START command to start a device or interface. Use the VARY TCPIP,,STOP command to stop a device or interface.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──┬─STArt─┬──┬─,device_name────┬──────────►◄
                     └─procname─┘     └─STOp──┘  └─,interface_name─┘
```

Parameters:

procname
The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* value is specified, the request will fail with an error message.

STArt
Start a device or interface known to TCP/IP.

STOp
Stop a device or interface known to TCP/IP.

device_name
The name of the device to be started or stopped.

interface_name
The name of the interface to be started or stopped.

Examples:
The following example shows how to start a device:

```
V TCPIP,,START,DEVD00
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,START,DEVD00
```

Usage:

- Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.STRTSTOP.
- When the VARY START command is used for XCF connection (specifying the CP name of the other node), the ISTLSXCF major node must be active on both nodes and the XCF TRLE for the connection must be active.

VARY TCPIP,,SYNTAXCHECK

Use the VARY TCPIP,,SYNTAXCHECK command to check the syntax of TCP/IP profile configuration statements without affecting the system operation or network configuration.

Subject to the restrictions listed below, you can direct this command to any TCP/IP stack that is of the same release level as the profile statements in the profile data set. The profile statements do not need to be related in any way to the active configuration.

Restriction: The `VARY TCPIP,,SYNTAXCHECK` command makes no attempt to update the active configuration; therefore, it does not detect and report conflicts with the active configuration. The following list shows some examples of the conflicts with the active configuration that `VARY TCPIP,,SYNTAXCHECK` cannot detect:

- Defining an interface more than once
- Deleting an interface that is not currently configured
- Defining an IPv6 interface on an IPv4-only stack

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──SYNTAXCHECK──,──datasetname──────────────►◄
                     └─procname─┘
```

Parameters:

procname
: The identifier of the TCP/IP address space. When the *procname* value is not specified, only one TCP/IP address space can be started. If more than one TCP/IP address space is available and no *procname* value is specified, the request fails with an error message.

SYNTAXCHECK
: Specify this parameter to check the syntax of profile statements without applying any changes to the system operation and network configuration. Put your profile statements in the data set specified by the *datasetname* value.

datasetname
: The *datasetname* value is required if you specify the SYNTAXCHECK parameter. The *datasetname* value is the name of a data set that contains TCP/IP profile configuration statements. The *datasetname* value must be a cataloged and fully-qualified data set name that is specified without any quotation marks. The *datasetname* value can be either a sequential data set or a member in a PDS.

Result: Syntax checking continues with any data set specified on an INCLUDE statement.

Restriction: The *datasetname* value cannot be the name of a z/OS UNIX file or a TCPIP.DATA data set.

Usage:

Guidelines

- If you have defined the resource profile MVS.VARY.TCPIP.SYNTAXCHECK in class OPERCMDS, authorize users to invoke the command by granting their user IDs CONTROL access to the resource profile.
- Issue this command to verify the statements in *datasetname* are free of syntax errors before activating the profile as the initial profile or with the VARY TCPIP,,OBEYFILE command.
- You do not need to direct this command to the TCP/IP stack that will use *datasetname* unless your profile contains MVS system symbols. If your profile contains MVS system symbols, you must direct this command to the TCP/IP stack that will activate *datasetname* for consistent resolution of the MVS system symbols.
- If your profile does not contain MVS system symbols, you can check the profile statement syntax by using any TCP/IP stack or host that supports the **VARY TCPIP,,SYNTAXCHECK** command, and that supports the statements in your profile. For consistent syntax checking, check the *datasetname* statement syntax by using a TCP/IP stack of the same z/OS release level as the TCP/IP stack that will activate *datasetname*.

Requirements

- Because syntax checking might stop for the current statement after a syntax error is detected, you must issue the command again after fixing any syntax errors reported by this command to ensure all syntax errors have been detected.
- For syntax checking consistency, you must run this command on a system of the same release level as the system that will activate the *datasetname* statements.

Results

- The profile is parsed and syntax errors are reported.
- No configuration changes are applied.
- No SMF events are generated.

The **VARY TCPIP,,SYNTAXCHECK** command makes no attempt to update the active configuration; therefore, it does not detect and report conflicts with the active configuration. The following list shows some examples of the conflicts with the active configuration that **VARY TCPIP,,SYNTAXCHECK** cannot detect:

- Defining an interface more than once
- Deleting an interface that is not currently configured
- Defining an IPv6 interface on an IPv4-only stack

VARY TCPIP,,SYSPLEX

Use the VARY TCPIP,,SYSPLEX command to change the sysplex configuration of the TCP/IP stack.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──────────────────────────────────────────►
                     └─procname─┘
```

```
►─SYSplex,──┬─LEAVEgroup──────────────────────────────────────────┬────────►◄
            ├─JOINgroup───────────────────────────────────────────┤
            ├─DEACTivate,DVIPA=dvipa──────────────────────────────┤
            ├─REACTivate,DVIPA=dvipa──────────────────────────────┤
            ├─QUIesce,POrt=portnum──┬───────────────────────────┬──┤
            │                       └─,JOBNAME=jobname──┬──────────┬┘  │
            │                                          └─,ASID=asid─┘   │
            ├─QUIesce,JOBNAME=jobname──┬─────────────┬────────────┤
            │                          └─,ASID=asid─┘             │
            ├─QUIesce,TARGET──────────────────────────────────────┤
            ├─RESUME,POrt=portnum──┬───────────────────────────┬───┤
            │                      └─,JOBNAME=jobname──┬──────────┬┘   │
            │                                         └─,ASID=asid─┘    │
            ├─RESUME,JOBNAME=jobname──┬─────────────┬─────────────┤
            │                         └─,ASID=asid─┘              │
            └─RESUME,TARGET───────────────────────────────────────┘
```

Parameters:

procname
The identifier of the TCP/IP address space. When the *procname* value is not specified, there can be only one TCP/IP address space started. If multiple TCP/IP address spaces are available and no *procname* value is specified, the request fails with an error message.

SYSplex
Requests to change a TCP/IP stack's DVIPA sysplex characteristics.

LEAVEgroup
Requests the TCP/IP stack to leave the sysplex group.

This causes the stack to leave the sysplex group, delete all dynamic DVIPAs, and deactivate all its configured VIPADYNAMIC definitions. The VIPADYNAMIC configuration information is retained for possible future use by the SYSPLEX,JOINGROUP command.

To rejoin the sysplex group it is necessary to issue a VARY TCPIP,,SYSPLEX,JOINGROUP operator command, which also reprocesses the stack's saved VIPADYNAMIC configuration.

Guideline: Do thisk only as a last resort if the operator has determined that this sysplex member is not functioning correctly and if the only other alternative would be to force the stack down. For more information, see sysplex problem detection and recovery information in the z/OS Communications Server: IP Configuration Guide.

Tip: The Netstat VIPADCFG/-F report can be used to view the saved VIPADYNAMIC configuration.

JOINgroup
Requests the TCP/IP stack to join the sysplex group.

When this command is issued, if VTAM is not running or if the DELAYJOIN parameter is configured for GLOBALCONFIG SYSPLEXMONITOR and OMPROUTE is not initialized, the join does not take place until after VTAM (and OMPROUTE, if DELAYJOIN is configured) is initialized. If this command is issued after the stack has left the sysplex group, it also reprocesses the stack's saved VIPADYNAMIC configuration.

If NOJOIN is configured on the GLOBALCONFIG SYSPLEXMONITOR statement, the JOINgroup command overrides NOJOIN. When you issue the JOINgroup command, the TCP/IP stack joins the sysplex group, as long as

VTAM is running and OMPROUTE is initialized (if the DELAYJOIN parameter is also configured for GLOBALCONFIG SYSPLEXMONITOR).

Tip: The Netstat VIPADCFG/-F report can be used to view the saved configuration prior to issuing the JOINgroup command.

Restriction: You cannot use this command to cause the stack to rejoin the sysplex group if the Sysplex Problem Detection cleanup function was unsuccessful and message EZZ9675E was issued, or if a previous attempt to process the saved VIPADYNAMIC configuration and join the TCP/IP sysplex group failed and message EZD1194 was issued. If either has occurred, you must restart the stack before it will be able to rejoin the sysplex group.

DEACTivate

Requests the TCP/IP stack to deactivate a dynamic VIPA. When you deactivate a dynamic VIPA, it appears as though the DVIPA has been deleted, but the DVIPA's configuration is saved.

DVIPA=*dvipa*

dvipa is the IPv4 address, IPv6 address, or IPv6 interface name of a dynamic VIPA (DVIPA) that is currently defined by VIPADEFINE or VIPABACKUP on this stack. The DVIPA can be in ACTIVE, BACKUP, or MOVING status.

The stack deactivates the DVIPA and ends any distribution for that DVIPA being done by this stack. The DVIPA configuration and any VIPADISTRIBUTE definitions are saved, and the deactivated DVIPA continues to be counted toward the maximum number of DVIPAs that can be defined on the stack. If there are existing connections to the DVIPA on this stack and there is another stack able to maintain the connections, the DVIPA is kept in QUIESCING status until the last connection terminates, and then the DVIPA is deactivated.

Guidelines:

- Deactivating an active DVIPA while the stack is part of the sysplex group allows an already-configured backup stack to takeover the DVIPA. (The stack that is serving as a backup for this DVIPA should have OMPROUTE active so that when it takes over the DVIPA it has the capability to advertise to others that it is the new owner).
- Deactivating a sysplex distributor DVIPA does not prevent the DVIPA from being marked as a target for distribution from another stack. As long as the application remains active on the stack, new connection requests can be distributed to it.
- Deactivating a backup DVIPA while the stack is part of the sysplex group makes the stack ineligible to takeover the DVIPA.
- This command can be issued after a stack has left the sysplex group. Because all the stack's DVIPA definitions are inactive while the stack is out of the group, the DVIPA is marked deactivated. If the stack later rejoins the group and restores its VIPADYNAMIC configuration, the DVIPA remains deactivated.
- A deactivated DVIPA can be reactivated using the VARY TCPIP,,SYSPLEX,REACTIVATE command.

Restriction: You cannot deactivate a VIPARANGE DVIPA created by BIND, SIOCSVIPA or SIOCSVIPA6 ioctl, or the MODDVIPA utility.

REACTivate
Requests that the TCP/IP stack redefine a deactivated dynamic VIPA using its saved configuration.

DVIPA=*dvipa*
dvipa is the IPv4 adddress, IPv6 address, or IPv6 interface name of a dynamic VIPA (DVIPA) that has been deactivated.

The stack will reestablish the DVIPA and any distribution for that DVIPA, based on the configuration that was saved when the DVIPA was deactivated.

Guidelines:

- Reactivating a VIPADEFINE DVIPA while the stack is part of the sysplex group allows a stack to take back the DVIPA.
- Reactivating a VIPABACKUP DVIPA while the stack is part of the sysplex group makes the stack again an eligible backup for the DVIPA, but does not typically trigger an immediate activation of the DVIPA. An exception to this behavior occurs when the following conditions are met:
  - The reactivated DVIPA's VIPABACKUP profile statement specified the MOVEABLE parameter.
  - The DVIPA is not active elsewhere in the sysplex.
- This command can be issued after a stack has left the sysplex group. Because all the stack's DVIPA definitions are inactive while the stack is out of the group, the DVIPA is marked as reactivated. If the stack later rejoins the group and restores its VIPADYNAMIC configuration, the DVIPA definition is restored.

QUIesce
Requests that the specified application, or all applications on a particular TCP/IP stack, be quiesced from DVIPA sysplex distributor workload balancing. After the command is issued, sysplex distributor will no longer route new TCP connection requests to the specified applications. Existing connections to these applications are not affected. This command must be issued on the local system where the applications are to be quiesced. This command can be useful in scenarios where you would like to temporarily divert new TCP connection requests away from a specific application or target system. One such scenario is when a particular application or system is to be shutdown (for example, in order to apply maintenance). Issuing this command prior to the shutdown can allow applications to gracefully complete any existing workload requests. PORT, JOBNAME or TARGET parameters must be specified following the QUIESCE keyword.

POrt=*portnum*
The port number parameter is an integer in the range 1 – 65 535 and is optional. Applications bound to this port number are excluded from DVIPA sysplex distributor workload balancing (they do not receive new TCP connection requests from sysplex distributor). If the *portnum* value specifies a port that has more than one instance of an application bound to it with either a different *jobname* or *asid* value, then either the JOBNAME value or the JOBNAME and ASID values must be specified to identify a unique specific application instance to be quiesced. PORT or TARGET parameters must be specified following the QUIESCE keyword.

JOBNAME=*jobname*
The *jobname* value specifies the MVS job name of the application with which the Quiesce command is associated.

- If the JOBNAME parameter is specified without the PORT keyword, then all applications with this *jobname* or *asid* value are quiesced regardless of the port they are bound to.
- If the *jobname* value specifies a job name that has more than one instance of an application with that job name but that has a different *asid* value, then the ASID parameter must also be specified and all application instances that have a matching job name are quiesced, regardless of the port they are using.
- The environment in which the application runs determines the job name that is to be associated with a particular client or server application.
- The *jobname* value can be up to 8 characters in length and is optional.

Guidelines:

- Applications submitted as batch jobs use the batch job name.
- Job names associated with applications started from the MVS operator console using the START command are determined as follows:
  - If the START command is issued with the name of a member in a cataloged procedure library (for example, S APP1), the job name is the member name (for example, APP1).
  - If the member name on the START command is qualified by a started task identifier (for example, S APP1.ABC), the job name is the started task identifier (for example, ABC).
  - The JOBNAME parameter can also be used on the START command to identify the job name (for example, S APP1,JOBNAME=XYZ).
  - The JOBNAME parameter can also be included on the JOB card.
- Applications run from a TSO user ID use the TSO user ID as the job name.
- Applications run from the z/OS shell normally have a job name that is the logged on user ID plus a one-character suffix.
- Authorized users can run applications from the z/OS shell and use the _BPX_JOBNAME environment variable to set the job name. In this case, the value specified for the environment variable is the job name.
- z/OS UNIX applications started by INETD typically use the job name of the INETD server plus a one-character suffix.

ASID=*asid*

The *asid* value is optional and specifies the hexadecimal address space ID associated with the application to be quiesced. If the *portnum* value specifies a port that has more than one instance of that application bound to it and the *jobname* value is not unique, then you can specify an *asid* value to quiesce all application instances that match this port, job name, and *asid* value.

Guidelines:

- This command must be issued on the system and the TCP/IP stack where the application instance is running.
- This command applies to a single TCP/IP stack's application instance. If the server needs to be quiesced over multiple stacks in a CINET environment, the command would need to be issued on each stack.
- Any sysplex distributor timed affinities will be terminated. Existing connections are not affected.

- The quiesce state is associated with the application's active listening socket. If the application is recycled or if the application closes and opens a new listening socket on the specified port, the socket will no longer be in a quiesced state.
- If the application is bound to the unspecified address, it can continue to receive connection requests that are not using a distributed DVIPA as the destination IP address.
- Applications quiesced with the PORT= option can be resumed by issuing a RESUME command.

Rule: When applications are quiesced using the PORT= or JOBNAME= option followed by a quiesce TARGET option for the stack on which those applications reside, you can no longer resume individual applications using the PORT= or JOBNAME= option. Instead, you must resume the entire TCP/IP stack using the TARGET option.

Tips:

- The Netstat ALL command can be issued as follows to determine which applications have been quiesced: QUIESCED DEST|NO.
- When an application is quiesced, the ready count (Rdy) field that appears on the Netstat VDPT display (issued on the sysplex distributor routing stack) is decremented. If no other applications are listening on this port on this target TCP/IP stack, the count is zero.

TARGET

Requests that all applications on this TCP/IP stack be quiesced from DVIPA sysplex distributor workload balancing. Existing connections are not affected.

Guidelines:

- This command must be issued on the system and the TCP/IP stack that is being quiesced.
- This command applies to a single TCP/IP stack. If an entire system with multiple TCP/IP stacks in the CINET environment needs to be quiesced, then a command needs to be issued for each TCP/IP stack on the system.
- Any sysplex distributor timer-based affinities are terminated. Existing connections are not affected.
- While sysplex distributor will no longer route new distributed DVIPA TCP connection requests to this TCP/IP stack, any TCP connections that do not specify a distributed DVIPA address as the destination IP address continue to be serviced by this TCP/IP stack.
- The QUIESCE state for a TARGET persists for all applications (existing and new) running on this TCP/IP stack, until the TCP/IP stack is recycled or a V TCPIP,,RESUME,TARGET command is issued.
- When an entire TCP/IP stack is quiesced using the TARGET option, you cannot resume individual applications for workload distribution. You can, however, resume distribution for the entire TCP/IP stack using the V TCPIP,,RESUME,TARGET command.
- When an entire TCP/IP stack is quiesced using the TARGET option, a quiesce for an individual application on that target stack is ignored.

Tips:

- The Netstat ALL command can be issued to determine which applications have been quiesced: QUIESCED DEST|NO

- When a TCP/IP stack is quiesced, the ready count (Rdy) field that appears on the Netstat VDPT display (issued on the sysplex distributor routing stack) will be zero for all entries associated with this target TCP/IP stack.

RESUME

Requests that the specified application or all applications associated with a TCP/IP stack be resumed for DVIPA sysplex distributor workload balancing (become eligible for new TCP connection requests). A PORT, JOBNAME or TARGET value must be specified following the RESUME keyword.

POrt=*portnum*

The *portnum* value is an integer in the range 1 – 65 535. Applications bound to this port number will be resumed for DVIPA sysplex distributor workload balancing. If the *portnum* value specifies a port that has more than one instance of an application bound to it, then either the JOBNAME value or the JOBNAME and ASID values must be specified to identify a unique specific application instance to be resumed. PORT or TARGET value must be specified following the RESUME keyword.

JOBNAME=*jobname*

The *jobname* value specifies the MVS job name of the application with which the resume command is associated.

- If the JOBNAME parameter is specified without the PORT keyword, then all applications with this *jobname* or *asid* value are resumed, regardless of the port they are bound to.
- If the *jobname* value specifies a job name that has more than one instance of an application with that job name but with a different *asid* value, then you must also specify the ASID parameter and all application instances that have a job name that matches are resumed regardless of port value.
- The environment in which the application runs determines the job name that is to be associated with a particular client or server application.
- The *jobname* value is optional and can be up to 8 characters in length.

Guidelines:

- Applications submitted as batch jobs use the batch job name.
- The job name associated with applications started from the MVS operator console using the START command will be determined as follows:
  - If the START command is issued with the name of a member in a cataloged procedure library (for example, S APP1), the job name will be the member name (for example, APP1).
  - If the member name on the START command is qualified by a started task identifier (for example, S APP1.ABC), the job name will be the started task identifier (for example, ABC).
  - The JOBNAME parameter can also be used on the START command to identify the job name (for example, S APP1,JOBNAME=XYZ).
  - The JOBNAME value can also be included on the JOB card.
- Applications run from a TSO user ID use the TSO user ID as the job name.

- Applications run from the z/OS shell normally have a job name that is a combination of the logged on user ID plus a one-character suffix.
- Authorized users can run applications from the z/OS shell and use the _BPX_JOBNAME environment variable to set the job name. In this case, the value specified for the environment variable is the job name.
- z/OS UNIX applications started by INETD typically use the job name of the INETD server plus a one-character suffix.

ASID=*asid*
The optional *asid* value defines the hexadecimal address space ID that is associated with the application to be quiesced. If the *portnum* value specifies a port that has more than one instance of an application bound to it and the job name is not unique, then you can specify an *asid* value to quiesce all application instances that match this *portnum*, *jobname*, and *asid* value.

TARGET
Requests that all applications on this TCP/IP stack be resumed for DVIPA sysplex distributor workload balancing. PORT or TARGET must be specified following the RESUME keyword.

Guidelines:

- This command must be issued on the stack that is quiesced or the stack where the quiesced application instance is running.
- This command applies to a single TCP/IP stack's application instance. If the server needs to be resumed over multiple stacks in a CINET environment, the command would need to be issued on each stack.
- RESUME with the TARGET option is the only valid command following a QUIESCE with the TARGET option command.

Examples:
To request a stack to delete all its dynamic VIPAs and leave the sysplex group:

```
VARY TCPIP,,SYSPLEX,LEAVEGROUP
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,LEAVEGROUP
EZZ0053I COMMAND SYSPLEX,LEAVEGROUP COMPLETED SUCCESSFULLY
```

To request a stack to join the sysplex group and restore its dynamic VIPAs:

```
VARY TCPIP,,SYSPLEX,JOINGROUP
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,JOINGROUP
EZD1178I THE VARY TCPIP,,SYSPLEX,JOINGROUP COMMAND WAS ACCEPTED
EZD1176I TCPCS HAS SUCCESSFULLY JOINED THE TCP/IP SYSPLEX GROUP
EZD1192I THE VIPADYNAMIC CONFIGURATION WAS SUCCESSFULLY RESTORED FOR stack_name
```

To request a stack to deactivate a dynamic VIPA and save its configuration:

```
VARY TCPIP,,SYSPLEX,DEACTIVATE,DVIPA=203.1.1.99
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,DEACTIVATE,DVIPA=203.1.1.99
EZD1197I THE VARY TCPIP,,SYSPLEX,DEACTIVATE,DVIPA COMMAND COMPLETED SUCCESSFULLY
```

To request a stack to restore a dynamic VIPA that had been deactivated:

```
VARY TCPIP,,SYSPLEX,REACTIVATE,DVIPA=203.1.1.99
EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,REACTIVATE,DVIPA=203.1.1.99
EZD1189I THE VARY TCPIP,,SYSPLEX,REACTIVATE,DVIPA COMMAND COMPLETED SUCCESSFULLY
```

To request a stack to quiesce for DVIPA sysplex distributor workload balancing, all instances of an application listening on port 500 with the same *jobname* and *asid* values:

```
  VARY TCPIP,,SYSPLEX,QUIESCE,PORT=500
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,QUIESCE,PORT=500
  EZZ0053I COMMAND SYSPLEX,QUIESCE COMPLETED SUCCESSFULLY
```

To request a stack to quiesce, for DVIPA sysplex distributor workload balancing, a specific shareport application instance:

```
  VARY TCPIP,,SYSPLEX,QUIESCE,PORT=23,JOBNAME=job1,ASID=71
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,QUIESCE,PORT=23,JOBNAME=JOB1,ASID=71
  EZZ0053I COMMAND SYSPLEX,QUIESCE COMPLETED SUCCESSFULLY
```

To request a stack to quiesce, for DVIPA sysplex distributor workload balancing, all application instances:

```
  VARY TCPIP,,SYSPLEX,QUIESCE,TARGET
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,QUIESCE,TARGET
  EZZ0053I COMMAND SYSPLEX,QUIESCE COMPLETED SUCCESSFULLY
```

To request a stack to quiesce, for DVIPA sysplex distributor workload balancing, all instances of an application with the same *jobname* and *asid* values regardless of port:

```
VARY TCPIP,,SYSPLEX,QUIESCE,JOBNAME=job2
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,QUIESCE,JOBNAME=JOB2
  EZZ0053I COMMAND SYSPLEX,QUIESCE COMPLETED SUCCESSFULLY
```

To request a stack to resume for DVIPA sysplex distributor workload balancing, all instances of an application listening on port 500 with the same *jobname* and *asid* values:

```
  VARY TCPIP,,SYSPLEX,RESUME,PORT=500
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,RESUME,PORT=500
  EZZ0053I COMMAND SYSPLEX,RESUME COMPLETED SUCCESSFULLY
```

To request a stack to resume, for DVIPA sysplex distributor workload balancing, a specific shareport application instance:

```
  VARY TCPIP,,SYSPLEX,RESUME,PORT=23,JOBNAME=job1,ASID=71
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,RESUME,PORT=23,JOBNAME=JOB1,ASID=71
  EZZ0053I COMMAND SYSPLEX,RESUME COMPLETED SUCCESSFULLY
```

To request a stack to resume, for DVIPA sysplex distributor workload balancing, all application instances:

```
  VARY TCPIP,,SYSPLEX,RESUME,TARGET
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,RESUME,TARGET
  EZZ0053I COMMAND SYSPLEX,RESUME COMPLETED SUCCESSFULLY
```

To request a stack to resume, for DVIPA sysplex distributor workload balancing, all instances of an application with the same *jobname* and *asid* values regardless of port:

```
VARY TCPIP,,SYSPLEX,RESUME,JOBNAME=job2
  EZZ0060I PROCESSING COMMAND: VARY TCPIP,,SYSPLEX,RESUME,JOBNAME=JOB2
  EZZ0053I COMMAND SYSPLEX,RESUME COMPLETED SUCCESSFULLY
```

Usage:
Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.SYSPLEX.

VARY command: TN3270E Telnet server address space

The functions listed in Table 12 on page 274 support the VARY TCPIP command when it is directed to a TN3270E Telnet server.

Table 12. TN3270E Telnet servers that support the MVS VARY TCPIP command

Function	Command
HELP	“VARY TCPIP,*tnproc*,HELP”
OBEYFILE	“VARY TCPIP,*tnproc*,OBEYFILE” on page 275
TELNET	“VARY TCPIP,*tnproc*,<TELNET>” on page 275
LUNS	“VARY TCPIP,*tnproc*,LUNS” on page 282

VARY TCPIP,*tnproc*,HELP

Use the VARY TCPIP,*tnproc*,HElp command from the MVS operator console to display the syntax of MVS operator Vary commands for the TN3270E Telnet server (Telnet).

Format:

```
►►──Vary ──TCPIP──,──tnproc──,HElp──┬──────────────────────────┬──────────────►◄
                                    ├─,Obeyfile────────────────┤
                                    ├─,Telnet──┬──────────────┬┤
                                    │          ├─,ABENDTRAP───┤│
                                    │          ├─,ACT─────────┤│
                                    │          ├─,DEBug───────┤│
                                    │          ├─,INACT───────┤│
                                    │          ├─,QUIesce─────┤│
                                    │          ├─,RESUME──────┤│
                                    │          └─,STOp────────┘│
                                    └─,LUNS────────────────────┘
```

Parameters:

Obeyfile
Shows help on the VARY OBEYFILE command.

Telnet
Shows the available options on the DISPLAY TELNET command.

ABENDTRAP
Shows help on the VARY TELNET,ABENDTRAP command.

ACT
Shows help on the VARY TELNET,ACT command.

DEBug
Shows help on the VARY TELNET,DEBUG command.

INACT
Shows help on the VARY TELNET,INACT command.

LUNS
Shows help on the VARY LUNS commands.

QUIesce
Shows help on the VARY TELNET,QUIESCE command.

RESUME
Shows help on the VARY TELNET,RESUME command.

STOp
Shows help on the VARY TELNET,STOP command.

Examples:

```
V TCPIP,TNSERV,HELP,ABENDTRAP

EZZ6123I V TCPIP,TNPROC<,TELNET>,ABENDTRAP,XMODNAME
<,XRCODE<,XINSTANCE>>
```

VARY TCPIP,*tnproc*,OBEYFILE

Use the VARY TCPIP,*tnproc*,OBEYFILE command to make dynamic changes to the TN3270E Telnet server (Telnet) configuration without stopping and restarting the Telnet address space.

See z/OS Communications Server: IP Configuration Guide for information about how different parameter updates take effect with Obeyfile processing.

Format:

```
►►──Vary ──TCPIP──,──tnproc──┬─Obeyfile,─────┬──┬─datasetname───────┬──────────────►◄
                             └─CMD=Obeyfile,─┘  └─DSN=──datasetname─┘
```

Parameters:

tnproc
: The member name of the cataloged procedure used to start the Telnet address space.

CMD=OBEYFILE or OBEYFILE
: Specify this parameter to make dynamic changes to Telnet configuration without stopping and restarting Telnet. These changes are in effect until the Telnet cataloged procedure is started again or until another VARY OBEYFILE overrides them. Put your changes in the data set that is specified by the *datasetname* value. You can maintain different data sets that contain a subset of the Telnet configuration statements and process them while Telnet is running.

DSN=*datasetname* **or** *datasetname*
: The *datasetname* value is required after specifying the OBEYFILE parameter. The *datasetname* value is the name of a data set containing Telnet configuration statements. The *datasetname* value must be a cataloged data set and specified as fully qualified without any quotation marks. The *datasetname* value can be either a sequential data set or a member in a PDS.

Examples:

The following example updates Telnet configuration information without stopping and restarting the Telnet address space. In this example a Telnet address space is started by the identifier TNSERV and the sequential data set USER99.TNSERV.OBEYFIL1 contains Telnet configuration statements:

```
VARY TCPIP,TCPPROC,CMD=OBEYFILE,DSN=USER99.TCPIP.OBEYFIL1
```

Usage:

- Users can be authorized to invoke the command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.TELNET.OBEYFILE.
- The DSN= parameter cannot be a z/OS UNIX file.

VARY TCPIP,*tnproc*,<TELNET>

Use the VARY TCPIP,*tnproc*,<TELNET> commands to control the TN3270E Telnet server (Telnet). For additional information about Telnet, see details about accessing remote hosts using Telnet in the z/OS Communications Server: IP Configuration Guide. You must specify the Telnet procedure name. Because Telnet no longer runs in the TCP/IP address space, the keyword TELNET can be omitted.

Format

```
►►──Vary ──TCPIP──,tnproc──┬─────────┬──┬─,ABENDTRAP─┬──────────────────────────►◄
                           └─,Telnet─┘  ├─,ACT───────┤
                                        ├─,DEBUG─────┤
                                        ├─,INACT─────┤
                                        ├─,QUIesce───┤
                                        └─,STOp──────┘
```

The IPv6 address format is accepted wherever an IP address is specified. The result might be no matches, but the IPv6 address format is always accepted.

The VARY TCPIP,*tnproc*,<TELNET> commands give the operator complete control over stopping and starting Telnet and allowing clients to connect. Using the VARY TCPIP,*tnproc*,<TELNET> commands, you can control the Telnet port and the LUs in the profile table. The combination of the STOP, QUIESCE, RESUME, and OBEYFILE commands gives the operator complete control over when to stop and start Telnet services and when to allow end users to connect. To help manage commands that are related to multiple ports, commands support a PORT keyword.

Tip: All parameters that are entered after these commands can be in any order.

Following provides details of the VARY TCPIP,*tnproc*,<TELNET> commands that can be used.

VARY ABENDTRAP command:
The VARY ABENDTRAP command provides abend dumps that are based on a return code being set in a given module.

Format:

```
►►──Vary ──TCPIP──,tnproc──┬─────────┬──,ABENDTRAP──,modname───────────────────►
                           └─,Telnet─┘

►──┬───────────────────────┬───────────────────────────────────────────────────►◄
   └─rcode──┬───────────┬──┘
            └─instance──┘
```

Parameters:

tnproc
: The member name of the cataloged procedure used to start the Telnet address space.

Telnet
: Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

ABENDTRAP
: The Abend Trap keyword.

modname
The exact module name, a partial name with an asterisk (*) at the far right, or just an *. The * is a wildcard.

rcode
The exact return code reported on an earlier EZZ6035I message. If *rcode* is not specified, any *rcode* in the module listed is considered a match. The rcode value is the left portion of the RCODE field on the EZZ6035I message. For example, if `RCODE: 3011-02` is presented, the rcode value is `3011` and the instance value is `02`.

instance
The exact instance reported on an earlier EZZ6035I message. To specify *instance*, *rcode* must also be specified. If *instance* is not specified, any instance is considered a match. The instance value is the right portion of the RCODE field on the EZZ6035I message. For example, if `RCODE: 3011-02` is presented, the instance value is `02` and the rcode value is `3011`.

Usage:

Module name, return code, or instance will be syntax checked. If an incorrect module name is used, the Abend Trap must be turned off and reset with the correct name. The same process is used if an incorrect return code or instance is used.

After the Abend Trap is set, it stays in effect until the trap is sprung or until it is turned off by issuing `V TCPIP,TN3270,TELNET,ABENDTRAP,OFF`. To change the trap, the current trap must first be turned off.

Authorization is through the user's RACF profile containing the MVS.VARY.TCPIP.TELNET.ABENDTRAP definition for ABENDTRAP. The definition can contain a wildcard at the TELNET or TCPIP level (for example MVS.VARY.TCPIP.**).

VARY ACT command:

The VARY ACT command changes the availability status of a VTAM LU for Telnet server usage. ACT enables the specified LU to be a candidate to represent a Telnet client.

Format:

```
►►──Vary TCPIP──,tnproc──┬─────────┬──,ACT──,luname────────────────────────►◄
                         └─,Telnet─┘
```

Parameters:

tnproc
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

ACT
The activate keyword.

`luname`
The name of the LU that you are activating. The LU name ALL has special meaning. It enables all deactivated LUs.

Usage:
The ACT command does not change the VTAM status of the LU. Use the INACTLUS display to show a list of LUs that are currently inactive.

Authorization is through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.ACT definition for ACT. The definition can contain a wildcard at the TELNET or TCPIP level (for example MVS.VARY.TCPIP.**).

VARY DEBUG command:

The VARY DEBUG command changes the DEBUG function on all active Telnet profiles.

Format:

```
►►──Vary TCPIP──,tnproc──┬─────────┬──,DEBug──,OFF──────────────────────►◄
                         └─,Telnet─┘
```

Parameters:

`tnproc`
The member name of the cataloged procedure used to start the Telnet address space.

`Telnet`
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

`DEBug`
The debug keyword.

`OFF`
All Telnet DEBUG functions are turned off for all active profiles.

Usage:
Authorization is through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.DEBUG definition for DEBUG. The definition can contain a wildcard at the TELNET or TCPIP level (for example MVS.VARY.TCPIP.**).

VARY INACT command:

Purpose:
The VARY INACT command changes the availability status of a VTAM LU for Telnet server usage. INACT disables the LU as a candidate to represent a Telnet client.

Format:

```
►►──Vary TCPIP──,tnproc──┬─────────┬──,INACT──,luname───────────────────►◄
                         └─,Telnet─┘
```

Parameters:

tnproc
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

INACT
The inactive keyword.

luname
The name of the LU you are deactivating.

Usage:

- The VARY INACT command does not change the VTAM status of the LU. Use the INACTLUS display to show a list of LUs currently inactive.
- If the specified LU has an active VTAM session, it is not affected by this command.
- Use the VTAM VARY NET,INACT command to end the SNA LU session.
- Use the TCP/IP VARY DROP command to end the TCP/IP connection.
- Authorization is through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.INACT definition for INACT. The definition can contain a wildcard at the TELNET or TCPIP level (for example MVS.VARY.TCPIP.**).

VARY QUIESCE command:

The VARY QUIESCE command removes the listener from the Telnet socket, which causes the specified port to not accept any new Telnet client connections. Currently established connections continue to be serviced.

Note: This command is not necessary for Obeyfile processing. An Obeyfile update creates a new profile for new connections but does not change configuration values for existing Telnet connections.

A qualified port cannot be specified. For information about qualified ports, see details about accessing remote hosts using Telnet in the z/OS Communications Server: IP Configuration Guide.

Format:

```
►►──Vary TCPIP──.tnproc──┬──────────┬──,QUIesce──┬─────────────────────┬──────────────►◄
                         └─,Telnet──┘            ├─,POrt=ALL───────────┤
                                                 ├─,POrt=num───────────┤
                                                 ├─,POrt=num1..num2────┤
                                                 ├─,POrt=Basic─────────┤
                                                 └─,POrt=Secure────────┘
```

Parameters:

tnproc
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

QUIesce
The QUIesce command keyword.

POrt=ALL|*num*|*num1..num2*|**Basic**|**Secure**
Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), basic ports, or secure ports should be quiesced.

- Using POrt=Basic selects all ports defined as BASIC (that is, TELNETPARMS contains a PORT statement).
- Using POrt=Secure selects all ports defined as SECURE (that is, TELNETPARMS contains a SECUREPORT or TTLSPORT statement).
- Port is optional if only one port is active; otherwise, a port option must be specified.

Usage:
Authorization is through the user's RACF profile containing the MVS.VARY.TCPIP.TELNET.QUIESCE definition for QUIESCE. The definition can contain a wildcard at the TELNET or TCPIP level (for example MVS.VARY.TCPIP.**).

VARY RESUME command:

The VARY RESUME command causes the currently quiesced port to begin accepting new Telnet client connections again using either the existing profile or a new profile.

Note: This command is not necessary for Obeyfile processing. An Obeyfile update creates a new profile for new connections but does not change configuration values for existing Telnet Connections.

A qualified port cannot be specified. For details about qualified ports, see the information about accessing remote hosts using Telnet in z/OS Communications Server: IP Configuration Guide.

Format:

```
►►──Vary TCPIP──,──tnproc──,──┬────────┬──,──RESUME──┬──────────────────┬──────────►◄
                              └─Telnet─┘             ├─,POrt=ALL────────┤
                                                     ├─,POrt=num────────┤
                                                     ├─,POrt=num1..num2─┤
                                                     ├─,POrt=Secure─────┤
                                                     └─,POrt=Basic──────┘
```

Parameters:

tnproc
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

RESUME
The RESUME keyword.

POrt=ALL|*num*|*num1..num2*|Basic|Secure
Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), basic ports, or secure ports should be quiesced.

- Using POrt=Basic selects all ports defined as BASIC (that is, TELNETPARMS contains a PORT statement).
- Using POrt=Secure selects all ports defined as SECURE (that is, TELNETPARMS contains a SECUREPORT or TTLSPORT statement).
- Port is optional if only one port is active; otherwise, a port option must be specified.

Usage:
Authorization is through the user's RACF profile containing the MVS.VARY.TCPIP.TELNET.RESUME definition for RESUME. The definition can contain a wildcard at the TELNET or TCPIP level (for example MVS.VARY.TCPIP.**).

VARY STOP command:

The VARY STOP command ends the port connection and all active connections. The STOP command does not end all of Telnet. The command processor remains active. You can issue a VARY OBEYFILE command to ACTIVATE a Telnet port using the Telnet configuration parameters.

Note: A qualified port cannot be specified. For details about qualified ports, see the information about accessing remote hosts using Telnet in z/OS Communications Server: IP Configuration Guide.

Format:

```
►►──Vary TCPIP──,tnproc──┬──────────┬──,STOp──┬───────────────────┬──────────────────►◄
                         └─,Telnet──┘         ├─,POrt=ALL─────────┤
                                              ├─,POrt=num─────────┤
                                              ├─,POrt=num1..num2──┤
                                              ├─,POrt=Secure──────┤
                                              └─,POrt=Basic───────┘
```

Parameters:

tnproc
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Legacy parameter that directs the command to the Telnet component when Telnet could run in the TCP/IP stack.

STOp
The STOP command keyword.

POrt=ALL|*num*|*num1..num2*|Basic|Secure
Specifies that **ALL** ports, a specific port (*num*), port number range (*num1..num2*), basic ports, or secure ports should be quiesced.

- Using POrt=Basic selects all ports defined as BASIC (that is, TELNETPARMS contains a PORT statement).

- Using POrt=Secure selects all ports defined as SECURE (that is, TELNETPARMS contains a SECUREPORT or TTLSPORT statement).
- If only one port is active and POrt is not specified, the command affects that one port; otherwise, POrt is required.
- Port is optional if only one port is active; otherwise, a port option must be specified.

Usage: Users can be authorized to invoke the STOP command by permitting their user IDs for CONTROL access to the RACF profile name MVS.VARY.TCPIP.TELNET.STOP. This profile name can contain a wildcard.

VARY TCPIP,*tnproc*,LUNS

Use the VARY TCPIP,*tnproc*,LUNS command to control the TN3270E Telnet LU name server (LUNS). For additional details about the Telnet LUNS, see the information about accessing remote hosts using Telnet in z/OS Communications Server: IP Configuration Guide.

Format:

```
                                                ┌─,MAX=100──┐
►►──VARY TCPIP──,tnproc──,LUNS──┬─,ACT─────┬────┼───────────┼──────────────────►◄
                                ├─,INACT───┤    └─,MAX=nn|*─┘
                                ├─,QUIesce─┤
                                ├─,RESUME──┤
                                └─,STArt───┘
```

The operator can use the VARY TCPIP, *tnproc*,LUNS commands to control the Telnet LUNS and its operational status and available LUs. The operator can use a combination of the START, QUIESCE, RESUME, ACT, and INACT commands to control when to start the LUNS, when to allow the LUNS to participate in recovery scenarios, when to allow Telnet LU name requesters (LUNRs) to connect, and which LUs are available for LUNRs to request.

Tip: You can enter all parameters for these commands in any order.

The following descriptions provide details of the VARY TCPIP,*tnproc*,TELNET,LUNS commands that you can issue.

VARY TCPIP,*tnproc*,LUNS,ACT command:

The VARY TCPIP,*tnproc*,LUNS,ACT command changes the availability status of a VTAM LU for LUNS usage. The ACT command enables the specified LU to be a candidate for request by a LUNR. The LU must also be active on the LUNR. The ACT command does not change the VTAM status of the LU. Use the D TCPIP,*tnproc*,LUNS,INACTLUS command to list the LUs that are currently inactive.

Format:

```
►►──Vary TCPIP──,tnproc──,LUNS──,ACT──,luname──────────────────────────────────►◄
```

Parameters:

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

LUNS
Indicates that the command is directed to a LUNS rather than to a local Telnet server.

ACT
The ACT command (activate) keyword.

luname
The name of the LU that you are activating. The LUNAME ALL keyword enables all inactivae LUs.

Usage:

Authorization is controlled through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.LUNS.ACT definition for the ACT command. The definition can contain a wildcard at the TELNET level (for example, MVS.VARY.TCPIP.TELNET.**).

VARY TCPIP,*tnproc*,LUNS,INACT command:

The VARY TCPIP,*tnproc*,LUNS,INACT command changes the availability status of a VTAM LU for LUNS usage. The INACT command disables the LU as a candidate to be requested by a LUNR. The VARY INACT command does not change the VTAM status of the LU. Use the D TCPIP,*tnproc*,LUNS,INACTLUS to show a list of LUs currently inactive. If the specified LU is currently in use by a LUNR, it is not affected by this command. Use the VTAM VARY NET,INACT command to end the SNA LU session.

Format:

```
►►──Vary TCPIP──,tnproc──,LUNS──,INACT──,luname────────────────────────────────►◄
```

Parameters:

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

LUNS
Indicates that the command is directed to a LUNS rather than to a local Telnet server.

INACT
The INACT command (inactivate) keyword.

luname
The name of the LU that you are deactivating.

Usage:

Authorization is controlled through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.LUNS.INACT definition for the INACT command. The definition can contain a wildcard at the TELNET level (for example MVS.VARY.TCPIP.TELNET.**).

VARY TCPIP,*tnproc*,LUNS,QUIesce command:

The VARY TCPIP,*tnproc*,LUNS,QUIESCE command causes an LU name server (LUNS) that is in the STANDBY and JOIN state to be ineligible to participate in recovery scenarios or to start when the VARY START command is issued. A LUNS must be in the QUIESCE state to make a configuration change using the VARY TCPIP,*tnproc*,OBEYFILE,DSN command.

Format:

```
►►──Vary TCPIP──,tnproc──,LUNS──,QUIesce─────────────────────────────────────►◄
```

Parameters:

tnproc
: The member name of the cataloged procedure that is used to start the Telnet address space.

LUNS
: Indicates that the command is directed to a LUNS rather than to a local Telnet server.

QUIesce
: The QUIesce command keyword.

Usage:

Authorization is controlled through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.LUNS.QUIESCE definition for the QUIESCE command. The definition can contain a wildcard at the TELNET level (for example, MVS.VARY.TCPIP.TELNET.**).

VARY TCPIP,*tnproc*,LUNS,RESUME command:

The VARY TCPIP,*tnproc*,LUNS,RESUME command causes the currently QUIESCED LU Name Server (LUNS) to be eligible to participate in recovery scenarios and start when the VARY START command is issued.

Format:

```
►►──Vary TCPIP──,tnproc──,LUNS──,RESUME──────────────────────────────────────►◄
```

Parameters:

tnproc
: The member name of the cataloged procedure that is used to start the Telnet address space.

LUNS
: Indicates that the command is directed to a LUNS rather than to a local Telnet server.

RESUME
: The RESUME command keyword.

Usage:

Authorization is controlled through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.LUNS.RESUME definition for the RESUME command. The definition can contain a wildcard at the TELNET level (for example MVS.VARY.TCPIP.TELNET.**).

VARY TCPIP,*tnproc*,LUNS,STArt command:

The VARY TCPIP,*tnproc*,LUNS,START command causes the current LU name server (LUNS) that is in standby mode to become active.

Format:

```
►►──Vary TCPIP──,tnproc──,LUNS──,STArt──────────────────────────────────►◄
```

Parameters:

tnproc
The member name of the cataloged procedure that is used to start the Telnet address space.

LUNS
Indicates that the command is directed to a LUNS rather than to a local Telnet server.

START
The START command keyword.

Usage:

Authorization is controlled through the user's RACF profile that contains the MVS.VARY.TCPIP.TELNET.LUNS.START definition for the START command. The definition can contain a wildcard at the TELNET level (for example, MVS.VARY.TCPIP.TELNET.**).

EZACMD command

Use the EZACMD command to issue selected z/OS Communications Server UNIX shell commands from the following environments:

- MVS console
- TSO
- NetView

EZACMD command installation

The EZACMD command is installed in the following target data sets:

- SYS1.SAXREXEC (placed in the System REXX system library for use with MVS console invocation)
- *hlq*.SEZAEXEC (for use with TSO or NetView invocations)

The following z/OS Communications Server UNIX shell commands are supported by EZACMD:

- **trmdstat** - see "The z/OS UNIX trmdstat command: Display traffic regulation management daemon (TRMD) log" on page 839 for details about command option syntax.

- **ipsec** - see "The z/OS UNIX ipsec command syntax" on page 700 for details about command option syntax.
- **nssctl** - see "The z/OS UNIX nssctl command syntax" on page 808 for details about command option syntax.
- **pasearch** - see "The z/OS UNIX pasearch command: Display policies" on page 819 for details about command option syntax.
- **ping** - see "The z/OS UNIX ping command: Send an echo request" on page 665 for details about command option syntax.

Restriction: For the EZACMD command, the **ping** command is supported only from the MVS console and NetView.

MVS console invocation of the EZACMD command

To use the EZACMD command from the MVS console, you must meet the following requirements:

- Verify that EZACMD was installed in the SYS1.SAXREXEC System REXX system library.
- The System REXX component must be configured and enabled on the z/OS image. See the System REXX information in z/OS MVS Programming: Authorized Assembler Services Guide and the AXR00 (default System REXX data set concatenation) information in z/OS MVS Initialization and Tuning Reference for details about enabling System REXX on your z/OS image.
- An operator must be logged on to the console (using the LOGON console command).
- The operator user ID must have a valid OMVS segment in the security database.

Guideline: The output from the command might not be formatted correctly as a result of the line width restriction for the MVS console. For the best formatting results, invoke the command from TSO, NetView, or the UNIX shell.

Rule: You must enclose the command and options in single quotation marks (') so that the MVS console support does not convert the command name and options to uppercase characters.

Format:

```
                                                          ┌─MAX=100───┐
►►──prefix──EZACMD──'command name──┬─────────────────┬──┼───────────┼──'──────────►◄
                                   └─command options─┘  └─MAX=lines─┘
```

Parameters

prefix
: The System REXX command prefix as defined in the AXR*nn* parmlib member. For example, if the prefix is defined as %% then invoke the command as %%EZACMD.

command name **and** *options*
: The command name is one of the supported z/OS Communications Server UNIX shell commands.

Command options are any options that the UNIX shell command supports. You must enter the command options in the exact format and case that is required by the z/OS UNIX command.

MAX= *lines*

MAX is an optional keyword that limits the number of output lines that are produced by the z/OS UNIX command and that are displayed on the MVS console. The default value is 100. The value can be any numeric value in the range 1 - 64 000.

Example 1:
Display currently active IPSec filters, limiting the number of output lines to 9.

```
%%EZACMD 'ipsec -f display max=9'
System REXX EZACMD: ipsec command - start - userID=USER1
System REXX EZACMD: ipsec -f display
CS V2R1 ipsec  Stack Name: TCPCS  Fri Oct  7 14:07:07 2011
Primary:  Filter          Function: Display           Format:   Detail
Source:   Stack Policy    Scope:    Current           TotAvail: 730
Logging:  On              Predecap: Off               DVIPSec:  No
NatKeepAlive:  0
Defensive Mode: Inactive

FilterName:                   DVIPA1 2
FilterNameExtension:          1
System REXX EZACMD: Maximum number of output lines (9) has been reached.
System REXX EZACMD: ipsec command - end - RC=4
```

Example 2:
Ping host w3.ibm.com:

```
%%EZACMD 'ping -v w3.ibm.com'
System REXX EZACMD: ping command - start - userID=USER1
System REXX EZACMD: ping -v w3.ibm.com
CS V2R1: Pinging host w3.ibm.com (9.17.137.11)
with 256 bytes of ICMP data
Ping #1 from 9.17.137.11: bytes=264 seq=1 ttl=242 time=62.48 ms
Ping #2 from 9.17.137.11: bytes=264 seq=2 ttl=242 time=61.98 ms
Ping #3 from 9.17.137.11: bytes=264 seq=3 ttl=242 time=86.88 ms
Ping statistics for w3.ibm.com (9.17.137.11)
  Packets: Sent=3, Received=3, Lost=0 (0% loss)
  Approximate round trip times in milliseconds:
  Minimum=61.98 ms, Maximum=86.88 ms, Average=70.45 ms, StdDev=14.23 ms
System REXX EZACMD: ping command - end - RC=0
```

TSO invocation of the EZACMD command

To use the EZACMD command in the TSO environment, you must meet the following requirements:

- Concatenate the *hlq*.SEZAEXEC library to the SYSEXEC or SYSPROC DD statements in your TSO logon procedure.
- The TSO user ID must have a valid corresponding OMVS segment defined in the security database.

TSO commands can be entered using ISPF option 6.

Format:

```
                                                     ┌─MAX=100───┐
►►──EZACMD──command name──┬──────────────────┬──┼───────────┼──────────────────────►◄
                          └─command options──┘  └─MAX=lines─┘
```

Parameters

command name **and** *options*

The command name is one of the supported z/OS Communications Server UNIX shell commands.

Command options are any options that the UNIX shell command supports. You must enter the command options in the exact format and case that is required by the z/OS UNIX command.

MAX= *lines*

MAX is an optional keyword that limits the number of output lines that are produced by the z/OS UNIX command and that are displayed for TSO. The default value is 100. The value can be any numeric value that is in the range 1 - 64 000.

Example 1:
Display currently active IPSec filters, limiting the number of output lines to 9.

```
===> EZACMD ipsec -f display max=9
TSO REXX EZACMD: ipsec command - start - userID=USER1
TSO REXX EZACMD: ipsec -f display

CS V2R1 ipsec  Stack Name: TCPCS  Fri Oct  7 14:30:02 2011
Primary:  Filter         Function: Display           Format:   Detail
Source:   Stack Policy    Scope:    Current           TotAvail: 730
Logging:  On             Predecap: Off               DVIPSec:  No
NatKeepAlive:  0
Defensive Mode: Inactive

FilterName:                  DVIPA1~2
FilterNameExtension:         1
TSO REXX EZACMD: Maximum number of output lines (9) has been reached.
TSO REXX EZACMD: ipsec command - end - RC=4
***
```

NetView invocation of the EZACMD command

To use the EZACMD command in the NetView environment, you must meet the following requirements:

- Concatenate the *hlq*.SEZAEXEC library to the DSICLD statement in your NetView logon procedure.
- Review the NetView security setup to understand which z/OS UNIX credentials are used. See the information about defining and verifying operator authority in *IBM Tivoli NetView for z/OS Administration Reference* for details.

Format:

```
                                                          ┌─MAX=100───┐
►►──NETVASIS EZACMD──command name──┬─────────────────┬───┼───────────┼──────────►◄
                                   └─command options─┘   └─MAX=lines─┘
```

Parameters

Command name and options

Command name is one of the supported z/OS Communications Server UNIX shell commands.

Command options are any option that the UNIX shell command supports. The command options must be entered in the exact format and case that is required by the z/OS UNIX command.

MAX= *lines*

MAX is an optional keyword that limits the number of output lines that are produced by the z/OS UNIX command and that are displayed for NetView. The default value is 100. The value can be any numeric value that is in the range 1 - 64 000.

Example 1:
Display currently active IPSec filters, limiting number of output lines to 9.

```
netvasis EZACMD ipsec -f display max=9
* CNM01    ezacmd ipsec -f display max=9
C CNM01    NetView REXX EZACMD: ipsec command - start - userID=USER1
C CNM01    NetView REXX EZACMD: ipsec -f display
C CNM01
C CNM01    CS V2R1 ipsec  Stack Name: TCPCS  Fri Oct  7 14:33:59 2011
C CNM01    Primary:  Filter          Function: Display            Format: Detail
C CNM01    Source:   Stack Policy    Scope:    Current            TotAvail: 730
C CNM01    Logging:  On              Predecap: Off                DVIPSec:  No
C CNM01    NatKeepAlive:  0
C CNM01    Defensive Mode: Inactive
C CNM01
C CNM01    FilterName:                   DVIPA1~2
C CNM01    FilterNameExtension:          1
C CNM01    NetView REXX EZACMD: Maximum number of output lines (9) has been reached.
C CNM01    NetView REXX EZACMD: ipsec command - end - RC=4
```

Example 2:
Ping host w3.ibm.com.

```
netvasis ezacmd ping -v w3.ibm.com
* CNM01    ezacmd ping -v w3.ibm.com
C CNM01    NetView REXX EZACMD: ping command - start - userID=USER1
C CNM01    NetView REXX EZACMD: ping -v w3.ibm.com
C CNM01    CS V2R1: Pinging host w3.ibm.com (9.17.137.11)
C CNM01    with 256 bytes of ICMP data
C CNM01    Ping #1 from 9.17.137.11: bytes=264 seq=1 ttl=242 time=69.24 ms
C CNM01    Ping #2 from 9.17.137.11: bytes=264 seq=2 ttl=242 time=65.91 ms
C CNM01    Ping #3 from 9.17.137.11: bytes=264 seq=3 ttl=242 time=67.90 ms
C CNM01    Ping statistics for w3.ibm.com (9.17.137.11)
C CNM01      Packets: Sent=3, Received=3, Lost=0 (0% loss)
C CNM01      Approximate round trip times in milliseconds:
C CNM01      Minimum=65.91 ms, Maximum=69.24 ms, Average=67.68 ms, StdDev=1.68 ms
C CNM01    NetView REXX EZACMD: ping command - end - RC=0
```

TSO commands

The following topics describe some of the system administrator TSO commands.

- “Using the SMSG interface”
- “MAKESITE command” on page 290
- “TESTSITE command” on page 294
- “HOMETEST command” on page 295
- “MVPXDIS command” on page 295

Using the SMSG interface

The TSO SMSG interface also allows you to change the characteristics of an active task. This is the general format of SMSG.

Format

```
►►──SMSG──procname──parameter──────────────────────────────────────────────►◄
```

Parameters

procname
The name of the member in a procedure library that was used to start the server or address space.

Note: The SMSG command works when issued from TSO and should not be issued from the operator console.

parameter
Any of the parameters that are valid for the server.

Usage

The following servers support the MVS SMSG command. Not all servers support the same parameters. You can find further descriptions of the supported parameters in the information for that server. See information about monitoring the status of SMTP using the SMSG command in the z/OS Communications Server: IP User's Guide and Commands for details about SMTP SMSG support. See TSO SMSG command—Monitoring the Status of LPD in the z/OS Communications Server: IP User's Guide and Commands for information about using the TSO SMSG command to provide an interactive interface to the LPD server.

Server/Addr Space	Supported Parameters
SMTP	DEBUG, EXPIRE, HELP, NODEBUG, NOTRACE, QUEUES, SHUTDOWN, STATS, TRACE
Remote Print Server (LPD)	PRINT WORK, TRACE OFF, TRACE ON

MAKESITE command

Use MAKESITE as a TSO command or in a batch job to generate new *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO data sets. The parameters are the same for either a TSO command or a batch job invocation of MAKESITE.

Tip: Use ETC.IPNODES (in the format etc/ipnodes) to define local hosts tables as the preferred alternative to MAKESITE. For more information, see Resolver configuration in the z/OS Communications Server: IPv6 Network and Application Design Guide, which discusses the use of IPNODES by the resolver to locate IPv4 and IPv6 addresses and site names.

Format

```
►►──MAKESITE──┬─────────────┬─,─┬──────────────────────────────┬─,──────────────►
              └─HLQ=──hlq───┘   └─MGMTclas=──management_class──┘

►──┬────────────────────────┬─,─┬──────────────────────────┬─,──────────────────►
   └─DATAclas=──data_class──┘   └─STORclas=──storage_class─┘

►──┬──────────────┬─,─┬──────────────────────────┬─────────────────────────────►◄
   └─Unit=──unit──┘   └─VOLser=──volume_serial───┘
```

Parameters

HLQ=*hlq*

The high-level qualifier of both the input and output data sets. The name specified is appended to the HOSTS.LOCAL, HOSTS.SITEINFO and HOSTS.ADDRINFO data set names.

Minimum abbreviation: HLQ=,
Maximum length: 29 characters

MGMTclas=*management_class*

The SMS-managed management class. MGMTCLAS is valid only in an SMS environment.

Minimum abbreviation: MGMT=
Maximum length: eight characters

DATAclas=*data_class*

The SMS-managed data class. DATACLAS is valid only in an SMS environment.

Minimum abbreviation: DATA=
Maximum length: eight characters

STORclas=*storage_class*

The SMS-managed storage class. STORCLAS is valid only in an SMS environment.

Minimum abbreviation: STOR=
Maximum length: eight characters

Unit=*unit*

An esoteric device name.

Minimum abbreviation: U=
Maximum length: eight characters

VOLser=*volume_serial*

Volume serial number.

Minimum abbreviation: VOL=
Maximum length: 6 characters

Usage

- The optional parameters can be in any order.
- Blanks are not allowed in the syntax.
- MAKESITE gets its input from *hlq*.HOSTS.LOCAL, where the HLQ is derived in this order:
  - HLQ parameter specified either with the command or in the batch job.
  - TSO user ID or the TSO PROFILE PREFIX, if it is different from the *userid*. In a batch job, *userid* can come from any of several sources depending on the environment. It can be the user ID of the user who submitted the batch job, or it can be the batch job name.
  - The value specified with the DATASETPREFIX statement in TCPIP.DATA.
  - System default.

The output data sets produced by MAKESITE are prefixed by either the HLQ parameter specified either on the command or batch job or the TSO user ID or TSO PROFILE PREFIX, if it is different from the *userid*.

- If any MAKESITE parameters are specified incorrectly, MAKESITE still executes using defaults (for example, for an incorrect *hlq*, the default is the active *userid* or *jobname*).
- Components that use the output from MAKESITE follow the standard naming conventions. If a DATASETPREFIX has been specified, it is used as the high-level qualifier for HOSTS.SITEINFO and HOSTS.ADDRINFO.

Examples

If your current active HLQ was TCPIP.MVSA, you would follow these steps to run MAKESITE and rename the output data sets.

1. Run MAKESITE with the appropriate parameters to generate 2 new data sets from the new *hlq*.HOSTS.LOCAL data set.

   As a TSO command, you might enter:

   ```
   MAKESITE HLQ=TCPIP.H0004,MGMT=M0001,VOLSER=STRG01,UNIT=SYSDA
   ```

   As a batch job, you might use this JCL:

   ```
   //MAKESITE JOB ,TIME=2,NOTIFY=USER7
   //*
   //BATCH  EXEC PGM=MAKESITE,REGION=8000K,
   //  PARM='VOLSER=STRG01,UNIT=SYSDA,HLQ=TCPIP.H0004,MGMT=M0001'
   //*
   //STEPLIB DD DISP=SHR,DSN=TCPIP.SEZALOAD
   //SYSPRINT  DD  SYSOUT=*,DCB=(LRECL=132,RECFM=FBA,BLKSIZE=3960)
   //SYSABEND  DD  SYSOUT=*
   //
   ```

   Note the following information:

   - This JCL is not shipped with TCP/IP.
   - The size of the parameter string is limited to 100 bytes.
   - Keywords in the parameter string can be abbreviated as shown in the MAKESITE syntax descriptions.
   - Region size varies according to your configuration. Make sure that the region size specified is valid for your configuration.

   This will create TCPIP.H004.HOSTS.SITEINFO and TCPIP.H0004.HOSTS.ADDRINFO.

2. Rename your existing HOSTS.SITEINFO and HOSTS.ADDRINFO data sets. These data sets are currently accessed by TCP/IP users on the system and should not be deleted while TCP/IP is running.

   For example, change TCPIP.MVSA.HOSTS.SITEINFO to TCPIP.MVSA.HOSTS.SITEOLD and TCPIP.MVSA.HOSTS.ADDRINFO to TCPIP.MVSA.HOSTS.ADDROLD.

3. Rename the new HOSTS.ADDRINFO and HOSTS.SITEINFO data sets to replace the old ones.

   For example, change TCPIP.H0004.HOSTS.SITEINFO to TCPIP.MVSA.HOSTS.SITEINFO and TCPIP.H0004.HOSTS.ADDRINFO to TCPIP.MVSA.HOSTS.ADDRINFO.

The following example shows the output when the MAKESITE command is run as a batch job. When the MAKESITE command is run as a TSO command, the report format is the same except that the message numbers are not displayed.

Tip: Messages with numbers that end with an "E" (for errors) or a "W" (for warning) are issued when an error occurs. Messages with numbers that end with an "I" are informational messages and do not represent a problem. Some of the messages in the following example reflect internal processing statistics.

```
EZA0549I                  S T A T I S T I C S
EZA0550I DATASET: USER40.HOSTS.LOCAL
EZA0551I      TOTAL LINES: 24
EZA0552W      BAD LINES: (SKIPPED) 0
EZA0553I      DUPLICATE NAMES: 0
EZA0554I      CONFLICTS IN FIRST 8 LETTERS: 0
EZA0555I      1 NETWORKS, 1 GATEWAYS, 4 HOSTS
EZA0556I DATASET: USER40.HOSTS.SITEINFO
EZA0557I      TABLE SIZE: 13
EZA0558I      TOTAL ENTRIES: 4
EZA0559I      DISTINCT NAMES: 5
EZA0560I      COLLISIONS: 1
EZA0561I      AVERAGE PROBES/NAME: 1.200
EZA0562I DATASET: USER40.HOSTS.ADDRINFO
EZA0563I      TABLE SIZE: 11
EZA0564I      TOTAL ENTRIES: 5
EZA0565I      COLLISIONS: 0
EZA0566I      NAMES DROPPED: 0
```

EZA0549I
Identifies the start of the MAKESITE statistics report.

EZA0550I
Displays the name of the HOSTS.LOCAL data set processed by the MAKESITE command. The indented lines following this message apply to the HOSTS.LOCAL data set.

EZA0551I
Displays the total number of lines in the HOSTS.LOCAL data set, including comment lines.

EZA0552W
Displays the number of lines in the HOSTS.LOCAL data set that were not processed because of syntax errors.

EZA0553I
Displays the number of duplicate names found in the HOSTS.LOCAL data set.

EZA0554I
Displays the number of potential conflicts. A potential conflict is detected if an address defined in the HOSTS.LOCAL data set maps to multiple names and the first 8 bytes of these names are the same.

EZA0555I
Displays the number of each record type in the HOSTS.LOCAL data set. Valid record types are NET, GATEWAY, and HOST. The number displayed for hosts includes the entry generated by the MAKESITE command for the loopback address.

EZA0556I
Displays the name of the HOSTS.SITEINFO data set. The indented lines following this message apply to the HOSTS.SITEINFO data set.

EZA0557I
Displays the number of table entries created in the HOSTS.SITEINFO data set.

EZA0558I
Displays the number of HOSTS.SITESINFO table entries used and shown as Total Entries in this report.

EZA0559I
Displays the number of names processed (excluding duplicates) and shown as Distinct Names in this report. There can be more Distinct Names than Total Entries if an address maps to more than one name.

EZA0560I
Displays the number of times a hash value was mapped to a slot that was already in use; this value is shown as Collisions in this report. This message is informational only and does not indicate a problem.

EZA0561I
Displays the result of the following calculation: 1 + (Collisions /Distinct Names).

EZA0562I
Displays the name of the HOSTS.ADDRINFO data set. The indented lines following this message apply to the HOSTS.ADDRINFO data set.

EZA0563I
Displays the number of table entries created in the HOSTS.ADDRINFO data set.

EZA0564I
Displays the number of HOSTS.SITESINFO table entries used.

EZA0565I
Displays the number of times a hash value was mapped to a slot that was already in use. This message is informational only and does not indicate a problem.

EZA0566I
Displays the number of names that were dropped because more than six names were mapped to a particular address.

Usage

After running the MAKESITE command, you can test the correctness of the *hlq*.HOSTS.ADDRINFO and *hlq*.HOSTS.SITEINFO data sets with the TESTSITE command.

TESTSITE command

Use TESTSITE to verify that the *hlq*.HOSTS.ADDRINFO and *hlq*.HOSTS.SITEINFO data sets can correctly resolve the name of a host, gateway, or net.

Requirement: VMCF must be started for the TSO TESTSITE command to be successful because the TSO TESTSITE command uses the Pascal socket API. If VMCF is not started, an ABEND0D6 can occur.

Format

```
►►──TESTSITE──────────────────────────────────────────────────────────►◄
```

Parameters

There are no parameters for this command.

Examples

To test your HOSTS data sets, enter:

```
TESTSITE
```

When prompted for a name, enter the host, gateway or net name you want to verify.

When you have checked all the names in question, enter `QUIT` and press `ENTER`.

Usage

TESTSITE gets its input from the *hlq*.HOSTS.ADDRINFO and *hlq*.HOSTS.SITEINFO data sets, where the HLQ is derived in this order:

- TSO user ID or the TSO PROFILE PREFIX, if it is different from the *userid*.
- The value specified with the DATASETPREFIX statement in PROFILE.TCPIP and TCPIP.DATA.
- System default.

HOMETEST command

Use HOMETEST to verify your host name and address configuration. See Verifying PROFILE.TCPIP and TCPIP.DATA using HOMETEST in the z/OS Communications Server: IP Configuration Guide for additional details about the use of the HOMETEST command.

Enter HOMETEST as a TSO command.

Requirement: VMCF must be started for the TSO HOMETEST command to be successful because the TSO HOMETEST command uses the Pascal socket API. If VMCF is not started, an ABEND0D6 can occur.

Restriction: When you use this function, the total number of IPv4 IP addresses that can be configured to the TCP/IP stack is limited to 255 IP addresses. This limitation of 255 IP addresses applies to all IPv4 IP addresses, including loopback and dynamic VIPA.

Format

```
►►──HOMETEST──────────────────────────────────────────────────────────►◄
```

Parameters

There are no parameters for this command.

MVPXDIS command

The MVPXDISP command can be used for debugging VMCF problems. See Diagnosing VMCF/IUCV problems with the MVPXDISP command in z/OS Communications Server: IP Diagnosis Guide for more information about this command.

UNIX command

The UNIX command **pwtokey** can be used for password security. See “Using the pwtokey facility” on page 973 for more information about this command

Chapter 2. Sending electronic mail using z/OS UNIX sendmail

z/OS UNIX sendmail provides enhanced SMTP support, integrating with the existing SMTP mail server system to enable you to send mail across the Internet. z/OS UNIX sendmail replaces SMTPPROC as the primary SMTP server. z/OS UNIX sendmail uses standard sendmail configuration and operation files. Consequently, you can simply use the existing mail user agent (MUA) interface to use z/OS UNIX sendmail.

For a comprehensive discussion of sendmail, see the industry-accepted document *sendmail* by O'Reilly & Associates, Inc.

For more information about sendmail see http://www.sendmail.org. For the features added after version 8.8.7, see *SENDMAIL INSTALLATION AND OPERATION GUIDE*, that can be found at http://www.sendmail.org/~ca/email/doc8.12/op.html.

z/OS UNIX sendmail commands

Command-line switches are command-line arguments that begin with a hyphen (-) and precede the list of recipients (if any). The forms for the command-line switches, where -*Y* is a single letter, are:

-*Y* Boolean switch

-*Yarg* Switch with argument

All switches are single letters. A complete list is shown in Table 13.

Table 13. Supported command-line sendmail switches

Switch	Version of sendmail	Description
-Ac	V8.12 and above	Use submit.cf
-Am	V8.12 and above	Use sendmail.cf
-b	All versions	Set operating mode
-ba	V8.9 and above	Go into ARPANET mode
-bD	V8.8 and above	Run as a daemon, but do not fork
-bd	All versions	Run as a daemon
-bH	V8.8 and above	Purge persistent host status
-bh	V8.8 and above	Print persistent host status
-bi	All versions	Initialize alias database
-bm	All versions	Be a mail sender
-bP	V8.12 and above	Print number of entries in the queue(s); available only with shared memory support.
-bp	All versions	Print the queue
-bs	All versions	Run SMTP on standard input
-bt	All versions	Rule testing mode
-bv	All versions	Verify: do not collect or deliver
-C	All versions	Location of configuration file

Table 13. Supported command-line sendmail switches (continued)

Switch	Version of sendmail	Description
-d	All versions	Enter debugging mode
-F	All versions	Set the sender's full name
-f	All versions	Set sender's address
-G	V8.12 and above	Relay (gateway) submission of a message
-hN	V8.9 and above	Set the hop count to N
-i	V8.9 and above	Ignore dots alone on lines by themselves in incoming messages
-L tag	V8.10 and above	Set the identifier used in syslog messages to the supplied tag
-N	V8.8 and above	Specify DSN NOTIFY information
-n	All versions	Do not do aliasing
-O	V8.7 and above	Set a multicharacter option
-o	All versions	Set a single-character option
-p	V8.1 and above	Set protocol and host
-q	All versions	Process saved messages in the queue at given intervals
-R	V8.8 and above	DSN what to return on a bounce
-t	All versions	Get recipients from message header
-V	V8.8 and above	Specify the ENVID string
-v	All versions	Run in verbose mode
-X	V8.2 and above	Log transactions

sendmail daemon commands

The following commands or symbolic links produce the same results as the corresponding sendmail command line arguments or switches (described in Table 13 on page 297).

Table 14. Supported command-line sendmail aliases

Name	Switch	Description
hoststat	-bh	Print persistent host status (V8.8 and above)
mailq	-bp	Print the queue contents
newaliases	-bi	Rebuild the *aliases* file
purgestat	-bH	Purge persistent host status (V8.8 and above)
smtpd	-bd	Run as a daemon

hoststat: Print persistent host status

Use `hoststat` to print the status of the last mail transaction with all remote hosts.

`hoststat` is identical to the z/OS UNIX `sendmail -bh` command.

The `hoststat` utility exits 0 on success, and >0 if an error occurs.

Format

```
►►──hoststat──┬──────┬────────────────────────────────────────────────────────►◄
              └─ -v─┘
```

Parameters

-v Prints verbose results. Normally the results are limited to 27 characters. Use the **-v** option to show results limited to 79 characters, thus providing more information.

Examples

In the following example, the previous connections to *there.ufoa.edu* and *books.ora.com* were successful. The status for *books.ora.com* is currently being updated. The asterisk (*) signifies that the file is locked. The host *prog.ammers.com* shows no status because connection to it could not be made. The last line in the example shows that the connection to *fbi.dc.gov* was refused by that host.

```
hoststat -v

 -------------- Hostname ------- How long ago ---------Results---------
 there.ufoa.edu                      00:00:51 250 PAA27153 Message acce
*books.ora.com                       07:43:39 250 GAA01255 Message acce
 prog.ammers.com                     06:55:08 No status available
 fbi.dc.gov                          03:28:53 Connection refused
```

For each host that has saved status, the following information is printed:

Hostname
Name of the host that z/OS UNIX sendmail was connected to. It might not be the host name specified for the recipient; it could be an MX record instead. If a message has multiple recipients, a separate status line is produced for each unique host that is tried. If this name is prefixed with an asterisk (*), the status file is locked and currently being updated.

How long ago
Shows how long ago this status record was updated. It is printed in the form: `DD+HH:MM:SS`. `DD` is the number of days. If the status was updated less than a day ago, the `DD+` is omitted. `HH` is hours, `MM` is minutes, and `SS` is seconds.

Results
Shows the results of the last connections attempt, failure, or success. If no reason was stored, the result prints as `No status available`. If a result was stored, it prints as `smtp msg`.

The `smtp` is the SMTP reply code. The `msg` is the text of the message generated by the other end or other program.

mailq: Print the mail queue

The `mailq` command prints a summary of the mail messages queued for future delivery.

The first line printed for each message shows the internal identifier used on this host for the message, the size of the message in bytes, the date and time the message was accepted into the queue, and the envelope sender of the message.

The second line shows the error message that caused this message to be retained in the queue; it will not be present if the message is being processed for the first time.

`mailq` is identical to z/OS UNIX `sendmail -bp` command.

The `mailq` utility exits with a code of 0 on success, and >0 if an error occurs.

Format

```
►►──mailq──┬────┬──────────────────────────────────────────────────────────────►◄
           └─-v─┘
```

Parameters

The available option is:

-v Print verbose information. This adds the priority of the message and a single character indicator (+ or blank) indicating whether a warning message has been sent on the first line of the message. Additionally, extra lines can be intermixed with the recipients indicating the controlling user information. This information shows who will own any programs that are executed on behalf of this message and the name of the alias this command expanded from, if any.

newaliases: Rebuild the database for the mail aliases file

The `newaliases` command rebuilds the random access database for the mail aliases file */etc/mail/aliases*. It must be run each time this file is changed in order for the change to take effect.

`newaliases` is identical to z/OS UNIX `sendmail -bi` command.

The `newaliases` utility exits with a code of 0 on success, and >0 if an error occurs.

Format

```
►►──newaliases─────────────────────────────────────────────────────────────────►◄
```

purgestat: Purge host status information

The `purgestat` command clears (purges) all the host-status information that was being saved under the HostStatusDirectory option directory. Clearing is done by removing all the directories under the HostStatusDirectory directory. The HostStatusDirectory directory is not removed.

`purgestat` is identical to z/OS UNIX `sendmail -bH` command.

The `purgestat` utility exits with a code of 0 on success, and >0 if an error occurs.

Format

```
►►──purgestat──────────────────────────────────────────────────────────────────►◄
```

smtpd: Run sendmail in the background as a daemon

The smtpd command causes sendmail to run in the background as a daemon, listening for incoming SMTP mail. This mode of operation is usually combined with the **-q** command-line switch, which causes sendmail to periodically process the queue.

smtpd is identical to z/OS UNIX sendmail -bd command.

Format

```
►►──smtpd──┬──────┬──────────────────────────────────────────────────►◄
           └─ -q──┘
```

Parameters

-q Processes saved messages in the queue at given intervals.

Using the mailstats command

The z/OS UNIX sendmail program provides the ability to gather information that can be used to produce valuable statistics. The StatusFile (S) option is used to specify a statistics file into which delivery agent statistics can be saved. The Mailstats program prints a summary of those statistics by printing the statistics file.

Mailstats command: Printing statistics

Use the Mailstats command to print the statistics contained in the statistics file.

Format

```
►►──mailstats──┬─ - C──<conf filename>─┬─────────────────────────────►◄
               ├─ - f──<stat filename>─┤
               ├─ - o──────────────────┤
               ├─ - p──────────────────┤
               └─ - P──────────────────┘
```

Parameters

-C *<conf filename>*
Specifies the name of the sendmail configuration file to be used to locate and analyze the z/OS UNIX sendmail statistics file. If not specified, /etc/mail/sendmail.cf is used as the default.

-f *<stat filename>*
Specifies the name of the z/OS UNIX sendmail statistics file to be analyzed. If not specified, the statistics file is located on the StatusFile (S) option specified in the z/OS UNIX sendmail configuration file.

-o Requests mailer names be omitted from the formatted output.

-p Specifies that output information is to be in program-readable mode and statistics are cleared. If both **-p** and **-P** are specified, the statistics file is cleared.

-P Specifies that output information is to be in program-readable mode and statistics are not cleared. If both **-p** and **-P** are specified, the statistics file is cleared.

Results

The following example shows the result of a MAILSTAT command.

```
Statistics from Sat Feb 15 12:51:09 2003
 M msgsfr bytes_from msgsto bytes_to msgsrej msgsdis Mailer
 =============================================================
 T   0         0K    0          0K  0       0
 C   0               0               0
```

The first line of output shows the time the statistics file was begun. The M column shows the index into the internal array of delivery agents, and the Mailer shows the symbolic name. The lines that follow show:

msgsfr
The number of messages and the total size in kilobytes of the messages received for each delivery agent.

msgsto
The number of messages and the total size in kilobytes of the messages sent for each delivery agent.

msgsrej
The number of message rejects by each mailer.

msgsdis
The number of message discards by each mailer.

The bottom line shows the totals.

Note: A delivery agent that has handled no traffic is excluded from the report.

Chapter 3. Monitoring the TCP/IP network

This information describes how to use the following TCP/IP commands to obtain information from the network.

- The TSO NETSTAT and z/OS UNIX **netstat**/**onetstat** commands provide information about the status of the network. See “Netstat.”
- The TSO PING and z/OS UNIX **ping**/**oping** commands determine the accessibility of a foreign node. See “Ping” on page 656.
- The TSO RPCINFO and z/OS UNIX **orpcinfo** commands display the servers that are using RPC binding protocol Version 2 that are registered and operational with any portmapper or rpcbind servers on your network. See “Rpcinfo” on page 674.
- The TSO TRACERTE and z/OS UNIX **traceroute**/**otracert** commands help debug network problems. See “Traceroute” on page 679.

Netstat

The TSO NETSTAT and z/OS UNIX **netstat**/**onetstat** commands provide the following information:

- Information about the status of the local host, including information about TCP/IP connections, network clients, gateways, and devices
- DNS cache information from the system-wide resolver
- The ability to drop connections for users who have the MVS.VARY.TCPIP.DROP statement defined in their RACF profile

As new functions are added to TCP/IP in the z/OS Communications Server, new information is also needed from the Netstat command in terms of new command options, new Netstat reports, or changes to existing Netstat reports. Any program that post processes output lines from the Netstat command and depends on the content of these output lines from the Netstat command should be reviewed and possibly modified when maintenance or a new release of z/OS is being installed. For every new release, see z/OS Summary of Message and Interface Changes for a summary of the changes to the following Netstat commands and their associated report output:

- The DISPLAY TCPIP,,NETSTAT operator command
- The TSO NETSTAT command
- The z/OS UNIX **netstat** command

TSO NETSTAT command output parsing considerations

No message identifiers are displayed in the output for TSO NETSTAT if the command is issued from an IPv6-enabled stack or if the command is issued from an IPv4-only stack but the request is for a long format display. If you have developed REXX programs that issue Netstat commands under TSO and parse the output lines based on message identifiers, you need to change those REXX programs to use some other token in the output lines to decide the format of the line you are trying to parse.

Here are some tips that might make the migration easier for you:

- Several Netstat reports display table entries such as the CONN report or the BYTEINFO report. If you are receiving Netstat output in LONG format, these table entries now take up more than one output line. The first line in a table entry always starts at position one in the line, and the remaining lines that belong to that same table entry start with an offset of two (position three). You can use that to determine which lines are the start of a table entry and which are follow-on lines that belong to that same table entry.
- For the non-table type of reports, depending on the report you are parsing and the pieces of information you are looking for, you need to identify the individual lines on some other token than the MSGID, such a LNKNAME or DEVNAME.

A small REXX program produced the output in the following example based on a NETSTAT DEVLINKS report:

```
Link/Intf name =LOOPBACK          Bytes in =12387      Bytes out =12387
Link/Intf name =VIPA1             Bytes in =0          Bytes out =0
Link/Intf name =LINKEE            Bytes in =0          Bytes out =0
Link/Intf name =TR1               Bytes in =110614     Bytes out =363744
Link/Intf name =VIPLC0A86501      Bytes in =0          Bytes out =0
Link/Intf name =VIPL092A689F      Bytes in =0          Bytes out =0
```

This output was produced with a REXX program that used MSGIDs to identify lines. The sample REXX program is shown in the following example:

```
/* REXX */
/* Requires PROFILE MSGID - uses MSGIDs to identify lines     */
netstr = 'DEVLINKS'
address TSO "NETSTAT "netstr" STACK"
n = queued()
if n > 0 then do x=1 to n
   i = (n-x)+1
   pull line.i
end
line.0 = n
do x=1 to line.0
   parse upper var line.x msgid t1 t2 t3 t4 .
   if msgid = 'EZZ2761I' then do                 /* MSGID EZZ2761I */
      interface = t2
   end
   if msgid = 'EZZ2820I' then do                 /* MSGID EZZ2820I */
      bytesin = t2
      bytesout = t4
      st1 = 'Link/Intf name ='||substr(interface,1,18)
      st1 = st1||' Bytes in ='||substr(bytesin,1,10)
      st1 = st1||' Bytes out ='||substr(bytesout,1,10)
      say st1
   end
end
exit
```

The exact same output can be produced using a modified REXX program that does not use MSGIDs but specific tokens in the Netstat report. In the following example, the only changes required are in the *parse* and *if* statements.

```
/* REXX */
/* Does not require MSGIDs, uses tokens to identify lines     */
/* This REXX works with z/OS V1R10                             */
netstr = 'DEVLINKS'
address TSO "NETSTAT "netstr" STACK"
n = queued()
if n > 0 then do x=1 to n
   i = (n-x)+1
   pull line.i
end
line.0 = n
```

```
do x =1 to line.0
   parse upper var line.x t1 t2 t3 t4 .
   if t1 = 'LNKNAME:' | t1 = 'INTFNAME:' then do
      interface = t2
   end
   if t1 = 'BYTESIN' then do
      bytesin = t3
   end
   if t1 = 'BYTESOUT' then do
      bytesout = t3
      st1 = 'Link/Intf name = '||substr(interface,1,18)
      st1 = st1||' Bytes in = '||substr(bytesin,1,10)
      st1 = st1||' Bytes out = '||substr(bytesout,1,10)
      say st1
   end
end
exit
```

Provide security product access to Netstat command

Controlling access to Netstat command can be added by using security product resources defined in the following table. You can define the following new security product resource names in the SERVAUTH class to control users' access to the TSO NETSTAT or UNIX shell **netstat** command options. See the sample EZARACF member for examples of the security product commands used to create the resource names. If the SERVAUTH class is not active or if security product resource name is not defined, access to the Netstat command will not be restricted.

Note: Take care with applications that might be invoking Netstat under the covers. If the Netstat security resource names are defined, the user IDs associated with applications invoking Netstat under the covers need to be permitted for READ access to the resource names.

Resource names in SERVAUTH class	Netstat options
EZB.NETSTAT.*mvsname*.*tcpprocname*.*	All Netstat options
EZB.NETSTAT.*mvsname*.*tcpprocname*.ALL	ALL / **-A**
EZB.NETSTAT.*mvsname*.*tcpprocname*.ALLCONN	ALLCONN / **-a**
EZB.NETSTAT.*mvsname*.*tcpprocname*.ARP	ARP / **-R**
EZB.NETSTAT.*mvsname*.*tcpprocname*.BYTEINFO	BYTEINFO / **-b**
EZB.NETSTAT.*mvsname*.*tcpprocname*.CACHINFO	CACHINFO / **-C**
EZB.NETSTAT.*mvsname*.*tcpprocname*.CLIENTS	CLIENTS / **-e**
EZB.NETSTAT.*mvsname*.*tcpprocname*.CONFIG	CONFIG / **-f**
EZB.NETSTAT.*mvsname*.*tcpprocname*.COnn	CONN / **-c**
EZB.NETSTAT.*mvsname*.*tcpprocname*.DEFADDRT	DEFADDRT/**-l**
EZB.NETSTAT.*mvsname*.*tcpprocname*.DEVLINKS	DEVLINKS / **-d**
EZB.NETSTAT.*mvsname*.*tcpprocname*.GATE	GATE / **-g**
EZB.NETSTAT.*mvsname*.*tcpprocname*.HOME	HOME / **-h**
EZB.NETSTAT.*mvsname*.*tcpprocname*.IDS	IDS / **-k**
EZB.NETSTAT.*mvsname*.*tcpprocname*.ND	ND / **-n**
EZB.NETSTAT.*mvsname*.*tcpprocname*.PORTLIST	PORTLIST / **-o**
EZB.NETSTAT.*mvsname*.*tcpprocname*.RESCACHE	RESCACHE / **-q**
EZB.NETSTAT.*mvsname*.*tcpprocname*.ROUTE	ROUTE / **-r**
EZB.NETSTAT.*mvsname*.*tcpprocname*.SLAP	SLAP / **-j**

Resource names in SERVAUTH class	**Netstat options**
EZB.NETSTAT.*mvsname.tcpprocname*.SOCKETS	SOCKETS / **-s**
EZB.NETSTAT.*mvsname.tcpprocname*.SRCIP	SRCIP / **-J**
EZB.NETSTAT.*mvsname.tcpprocname*.STATS	STATS / **-S**
EZB.NETSTAT.*mvsname.tcpprocname*.TELNET	TELNET / **-t**
EZB.NETSTAT.*mvsname.tcpprocname*.TTLS	TTLS / **-x**
EZB.NETSTAT.*mvsname.tcpprocname*.UP	Up / **-u**
EZB.NETSTAT.*mvsname.tcpprocname*.VCRT	VCRT / **-V**
EZB.NETSTAT.*mvsname.tcpprocname*.VDPT	VDPT / **-O**
EZB.NETSTAT.*mvsname.tcpprocname*.VIPADCFG	VIPADCFG / **-F**
EZB.NETSTAT.*mvsname.tcpprocname*.VIPADYN	VIPADYN / **-v**

You can use the control statements in the sample JCL job provided in SEZAINST(EZARACF) to define these authorizations.

- If this is the first SERVAUTH class profile that your installation is using, activate the SERVAUTH class using the following commands:

  ```
  SETROPTS CLASSACT(SERVAUTH)
  SETROPTS RACLIST(SERVAUTH)
  ```

- **Example 1**: If you wanted to permit USER2 access to the Netstat CONN/**-c** option for TCP/IP stack TCP1 on system MVSA you could use the following definitions:

  ```
  RDEFINE SERVAUTH (EZB.NETSTAT.MVSA.TCP1.CONN) UACC(NONE)
  PERMIT (EZB.NETSTAT.MVSA.TCP1.CONN) ACCESS(READ) CLASS(SERVAUTH) ID(USER2)
  ```

- **Example 2**: If you wanted to permit USER4 to have access to all of Netstat options you could use the following definitions:

  ```
  SETROPTS GENERIC(SERVAUTH)
  RDEFINE SERVAUTH (EZB.NETSTAT.MVSA.TCP1.*) UACC(NONE)
  PERMIT (EZB.NETSTAT.MVSA.TCP1.*) ACCESS(READ) CLASS(SERVAUTH) ID(USER4)
  SETROPTS GENERIC(SERVAUTH) REFRESH
  ```

- Refresh RACLIST

  ```
   SETROPTS RACLIST(SERVAUTH) REFRESH
  ```

The TSO NETSTAT command syntax

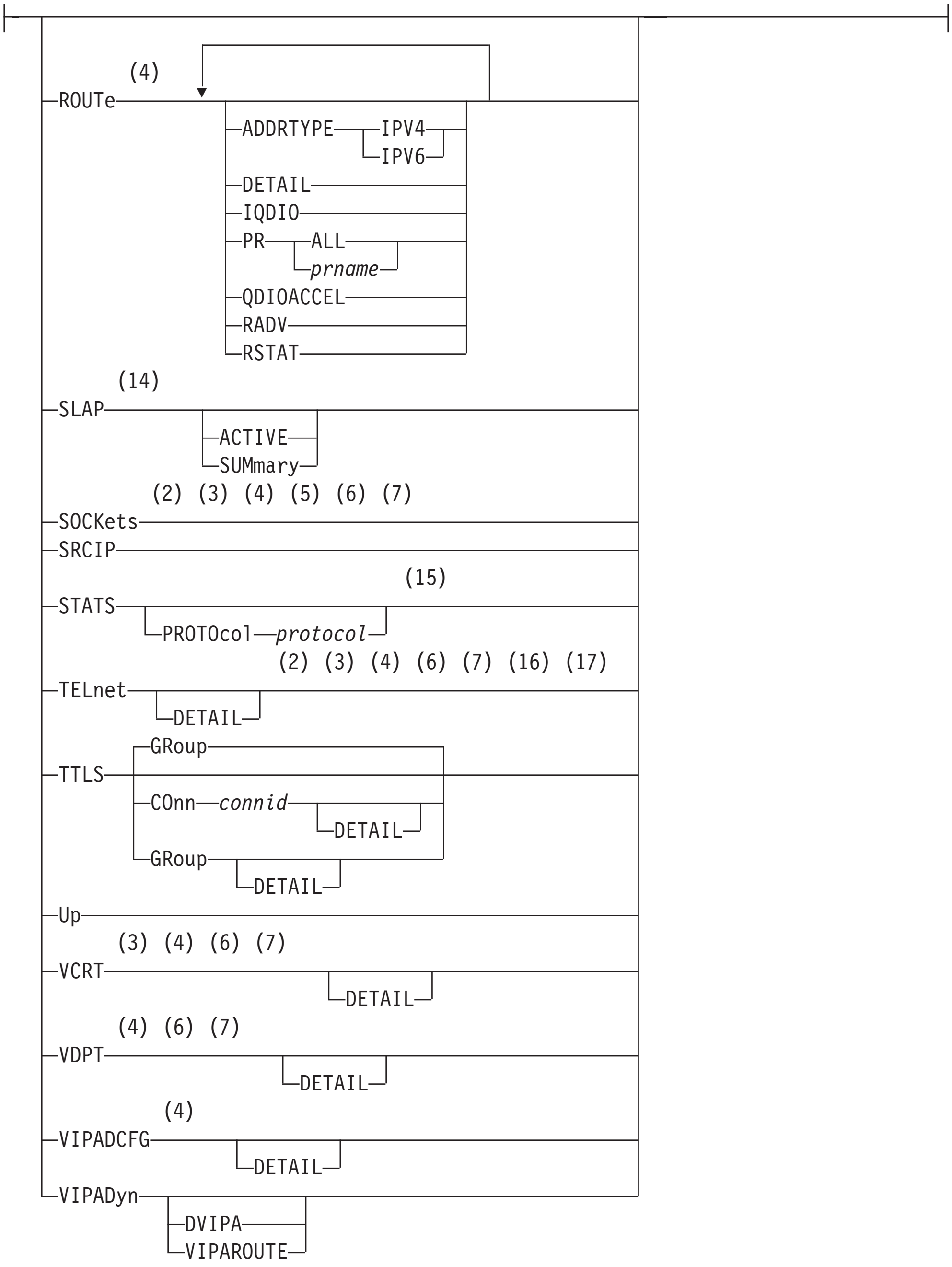
Use the TSO NETSTAT command to display the configuration and network status on a local TCP/IP stack.

Syntax

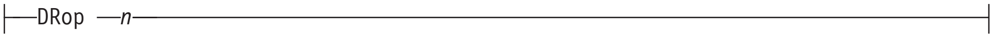
```
             (1)
►►──NETSTAT──────┬─┤ Report Option ├─┤ Target ├─┤ Output ├─┤ (Filter ├─┬──────►◄
                 └─┤ Command ├─┤ Target ├───────────────────────────────┘
```

Report Option:

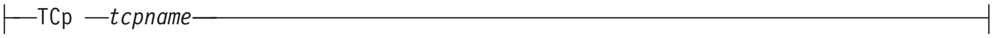
```
   ┌─COnn────────────────────────────────────────────────────────────┐
├──┼─────────────────────────────────────────────────────────────────┼──────────┤
   │                (2) (3) (4) (5) (6) (7) (8) (9)                  │
   ├─ALL─────────────────────────────────────────────────────────────┤
   │       └─SERVER─┘                                                │
   │                      (2) (3) (4) (5) (6) (7) (8) (9) (10)       │
   ├─ALLConn─────────────────────────────────────────────────────────┤
   │           └─APPLDATA─┘                                          │
   ├─ARp ──┬─net address─┬───────────────────────────────────────────┤
   │       └─ALL─────────┘                                           │
   │         (2) (3) (4) (5)                                         │
   ├─BYTEinfo────────────────────────────────────────────────────────┤
   │                               └─IDLETIME─┘                      │
   ├─CACHinfo────────────────────────────────────────────────────────┤
   │        (2) (5)                                                  │
   ├─CLients─────────────────────────────────────────────────────────┤
   ├─CONFIG──────────────────────────────────────────────────────────┤
   │        ┌───────────────┐                                        │
   │        ▼               │ (2) (3) (4) (5) (6) (7) (8) (9) (10)   │
   ├─COnn───────────────────┴────────────────────────────────────────┤
   │           ├─APPLDATA─┤                                          │
   │           └─SERVER───┘                                          │
   ├─DEFADDRT────────────────────────────────────────────────────────┤
   │                    (9) (11)                                     │
   ├─DEvlinks────────────────────────────────────────────────────────┤
   │             └─SMC─┘                                             │
   │       (4)                                                       │
   ├─Gate────────────────────────────────────────────────────────────┤
   │              └─DETAIL─┘                                         │
   ├─┬─HElp─┬────────────────────────────────────────────────────────┤
   │ └─?────┘                                                        │
   │       (11)                                                      │
   ├─HOme────────────────────────────────────────────────────────────┤
   │                        (12)                                     │
   ├─IDS─────────────────────────────────────────────────────────────┤
   │      ├─SUMmary────────────────┤                                 │
   │      └─PROTOcol──protocol─────┘                                 │
   │     (4)                                                         │
   ├─ND──────────────────────────────────────────────────────────────┤
   │         (6)                                                     │
   ├─PORTList────────────────────────────────────────────────────────┤
   │              ┌─SUMmary─────────────┐ (3) (4) (13)               │
   └─RESCache─────┼─────────────────────┼────────────────────────────┘
                  ├─DETAIL──────────────┤
                  │        └─NEGative─┘ │
                  └─SUMmary─────────────┘
                            └─DNS─┘
```



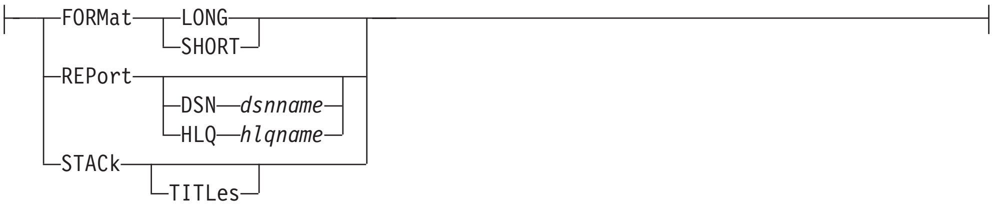
Command:



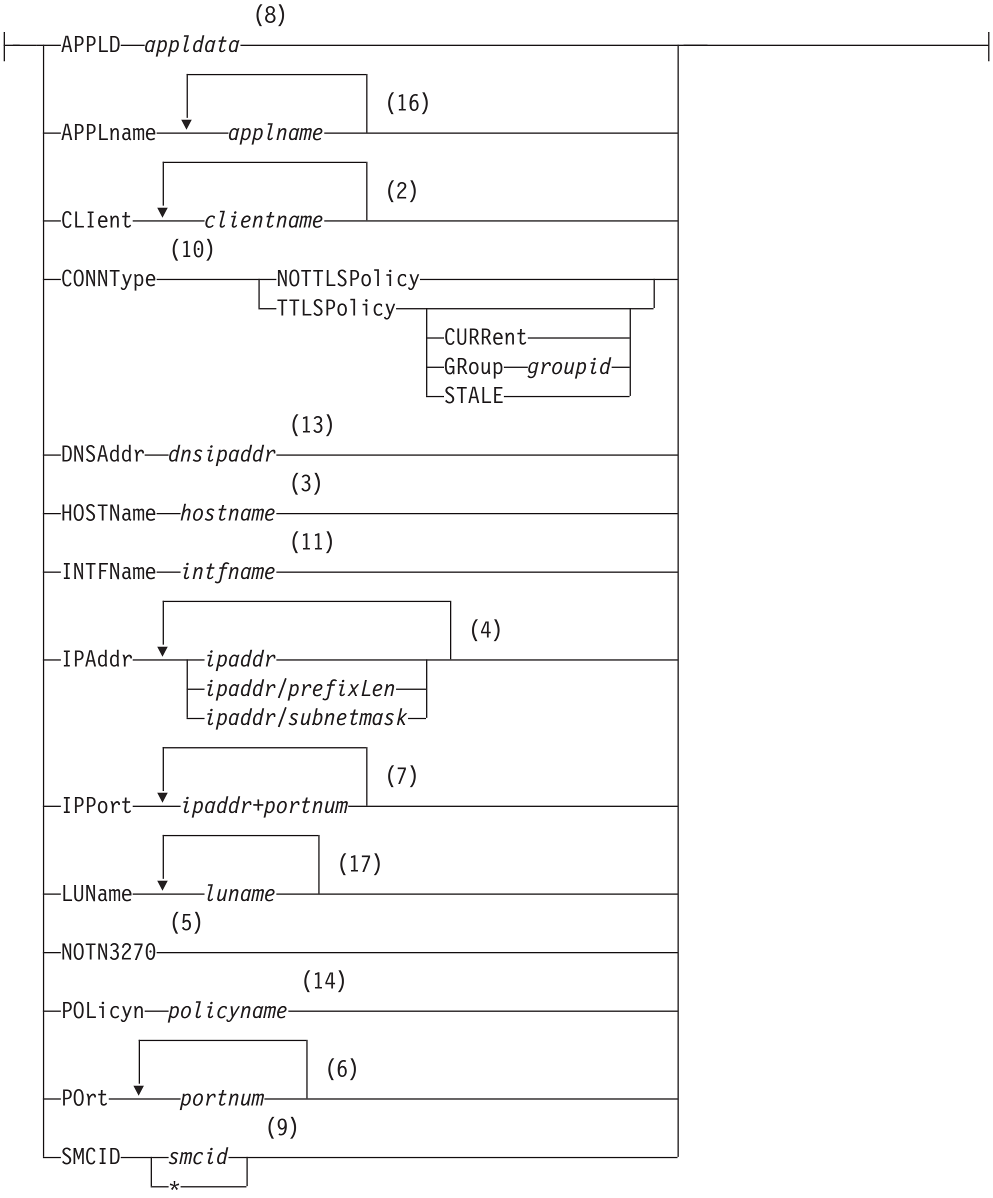
Target:



Output:



Filter:



Notes:

1 The minimum abbreviation for each parameter is shown in uppercase letters.

2 The CLIent filter is valid with ALL, ALLConn, BYTEinfo, COnn, CLients, SOCKets, and TELnet.

3 The HOSTName filter is valid only with ALL, ALLConn, BYTEinfo, COnn, RESCache, SOCKets, TELnet, and VCRT.

4 The IPAddr filter is valid only with ALL, ALLConn, BYTEinfo, COnn, Gate, ND, RESCache, ROUTe, SOCKets, TELnet, VCRT, and VDPT, and VIPADCFG.

5 The NOTN3270 filter is valid only with ALL, ALLConn, BYTEinfo, COnn, CLients, and SOCKets.

6 The POrt filter is valid only with ALL, ALLConn, COnn, PORTList, SOCKets, TELnet, VCRT, and VDPT.

7 The IPPort filter is valid only with ALL, ALLConn, COnn, SOCKets, TELnet, VCRT, and VDPT.

8 The APPLD filter is valid only with ALL, ALLConn, and COnn.

9 The SMCID filter is valid only with ALL, ALLConn, COnn, and DEvlinks.

10 The CONNType filter is valid only with ALLConn and COnn.

11 The INTFName filter is valid only with DEvlinks and HOme.

12 The valid protocol values are TCP and UDP.

13 The DNSAddr filter is valid only with RESCache.

14 The POLicyn filter is valid only with SLAP.

15 The valid protocol values are IP, ICMP, TCP, and UDP.

16 The APPLname filter is valid only with TELnet.

17 The LUName filter is valid only with TELnet.

The z/OS UNIX netstat command syntax

Use the z/OS UNIX **netstat** command to display the network configuration and status on a local TCP/IP stack.

Note:

1. **netstat** is a synonym for the **onetstat** command in the z/OS UNIX shell. The **onetstat** command syntax is the same as that for the **netstat** command.
2. Some option modifiers for the z/OS UNIX **netstat** command are shown below using uppercase letters followed by lowercase letters (for example, SUMmary). The portion of the modifier shown using uppercase letters indicates the minimum abbreviation for the modifier. The modifier used must be entered using all uppercase letters.

Syntax

```
►►──netstat──┬─┤ Report Option ├─┤ Target ├─┤ Output ├─┤ Filter ├─┬──────────►◄
             └─┤ Command ├─┤ Target ├─────────────────────────────┘
```

Report Option:

```
   ┌─ -c ────────────────────────────────────────────────────┐
├──┼─────────────────────────────────────────────────────────┼──────────┤
   │              (1) (2) (3) (4) (5) (6) (7) (8)            │
   ├─ -A──┬────────┬─────────────────────────────────────────┤
   │      └─SERVER─┘                                         │
   │                (1) (2) (3) (4) (5) (6) (7) (8) (9)      │
   ├─ -a──┬──────────┬───────────────────────────────────────┤
   │      └─APPLDATA─┘                                       │
   │                (2) (3) (4) (6)                          │
   ├─ -b──┬──────────┬───────────────────────────────────────┤
   │      └─IDLETIME─┘                                       │
   ├─ -C─────────────────────────────────────────────────────┤
   │      ┌──────────────┐                                   │
   │      ▼              │ (1) (2) (3) (4) (5) (6) (7) (8) (9)
   ├─ -c──┴─┬──────────┬─┴───────────────────────────────────┤
   │        ├─APPLDATA─┤                                     │
   │        └─SERVER───┘                                     │
   │           (8) (10)                                      │
   ├─ -d──┬─────┬────────────────────────────────────────────┤
   │      └─SMC─┘                                            │
   │   (2) (6)                                               │
   ├─ -e─────────────────────────────────────────────────────┤
   │              (4)                                        │
   ├─ -F──┬────────┬─────────────────────────────────────────┤
   │      └─DETAIL─┘                                         │
   ├─ -f─────────────────────────────────────────────────────┤
   │              (4)                                        │
   ├─ -g──┬────────┬─────────────────────────────────────────┤
   │      └─DETAIL─┘                                         │
   │   (10)                                                  │
   ├─ -h─────────────────────────────────────────────────────┤
   ├─ -J─────────────────────────────────────────────────────┤
   │               (11)                                      │
   ├─ -j──┬─────────┬────────────────────────────────────────┤
   │      ├─ACTIVE──┤                                        │
   │      └─SUMmary─┘                                        │
   │                          (12)                           │
   ├─ -k──┬──────────────────────┬───────────────────────────┤
   │      ├─SUMmary──────────────┤                           │
   │      └─PROTOcol──protocol───┘                           │
   ├─ -l─────────────────────────────────────────────────────┤
   │   (4)                                                   │
   ├─ -n─────────────────────────────────────────────────────┤
   │              (1) (4) (5)                                │
   ├─ -O──┬────────┬─────────────────────────────────────────┤
   │      └─DETAIL─┘                                         │
   │   (5)                                                   │
   └─ -o─────────────────────────────────────────────────────┘
```

```
├──┬────────────────────────────────────────────────────┬──────────────────────────────┤
   │           ┌─SUMmary────────────────┐ (3) (4) (13)   │
   ├─ -q───────┼────────────────────────┼────────────────┤
   │           ├─DETAIL──┬───────────┬──┤                │
   │           │         └─NEGative──┘  │                │
   │           └─SUMmary──┬───────┬─────┘                │
   │                      └─DNS───┘                      │
   ├─ -R──┬─net address─┬────────────────────────────────┤
   │      └─ALL─────────┘                                │
   │           ┌──────────────────────────┐               │
   │           ▼                          │ (4)           │
   ├─ -r───────┬─ADDRTYPE──┬─IPV4─┬───────┼───────────────┤
   │           │           └─IPV6─┘       │               │
   │           ├─DETAIL───────────────────┤               │
   │           ├─IQDIO────────────────────┤               │
   │           ├─PR───┬─ALL────┬──────────┤               │
   │           │      └─prname─┘          │               │
   │           ├─QDIOACCEL────────────────┤               │
   │           ├─RADV─────────────────────┤               │
   │           └─RSTAT────────────────────┘               │
   │                                   (14)               │
   ├─ -S───┬──────────────────────────┬──────────────────┤
   │       └─PROTOcol──protocol───────┘                   │
   │     (1) (2) (3) (4) (5) (6)                          │
   ├─ -s─────────────────────────────────────────────────┤
   │                 (1) (2) (3) (4) (5) (15) (16)        │
   ├─ -t───┬────────────┬────────────────────────────────┤
   │       └─DETAIL─────┘                                 │
   ├─ -u─────────────────────────────────────────────────┤
   │                 (1) (3) (4) (5)                      │
   ├─ -V───┬────────────┬────────────────────────────────┤
   │       └─DETAIL─────┘                                 │
   ├─ -v──┬──────────────┬───────────────────────────────┤
   │      ├─DVIPA────────┤                                │
   │      └─VIPAROUTE────┘                                │
   │       ┌─GRoup──────────────────────────┐              │
   ├─ -x───┼────────────────────────────────┼─────────────┤
   │       ├─COnn──connid──┬────────────┬───┤              │
   │       │               └─DETAIL─────┘   │              │
   │       └─GRoup──┬────────────┬──────────┘              │
   │                └─DETAIL─────┘                         │
   └─ -?─────────────────────────────────────────────────┘
```

Command:

```
├── -D── n──────────────────────────────────────────────────────────────────┤
```

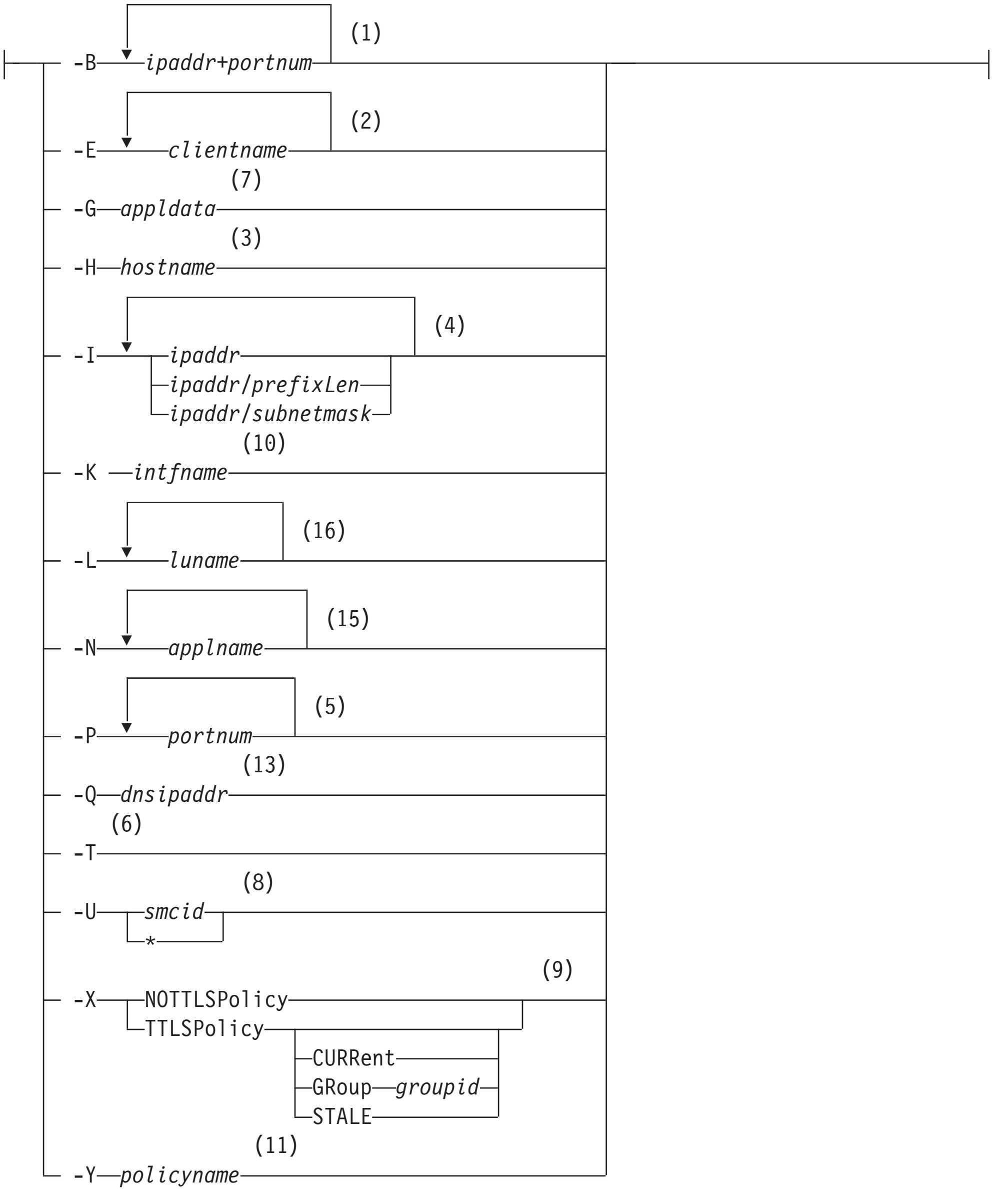
Target:

```
├── -p ──tcpname────────────────────────────────────────────────────────────┤
```

Output:

```
├── -M──┬─LONG──┬───────────────────────────────────────────────────────────┤
        └─SHORT─┘
```

Filter:



Notes:

1 -B filter is valid only with -A, -a, -c, -s, -t, -O, and -V.

2 -E filter is valid only with -A, -a, -b, -c, -e, -s, and -t.

3 -H filter is valid only with -A, -a, -b, -c, -q, -s, -t, and -V.

4 -I filter is valid only with -A, -a, -b, -c, -F, -g, -n, -O, -q, -r, -s, -t, and -V.

5 -P filter is valid only with -A, -a, -c, -O, -o, -s, -t, and -V.

6 -T filter is valid only with -A, -a, -b, -c, -e, and -s.

7 -G filter is valid only with -A, -a, and -c.

8 -U filter is valid only with -A, -a, -c and -d.

9 -X filter is valid only with -a, and -c.

10 -K filter is valid only with -d and -h.

11 -Y filter is valid only with -j.

12 The valid protocol values are TCP, and UDP.

13 -Q filter is valid only with -q.

14 The valid protocol values are ICMP, IP, TCP, and UDP.

15 -N filter is valid only with -t.

16 -L filter is valid only with -t.

The Netstat parameter overview

The following describes the individual parameter topics that are identified in the syntax diagram. The parameter format that is used below is the TSO parameter keyword followed by a slash and the z/OS UNIX shell character parameter. If a TSO parameter is not followed by a slash and a z/OS UNIX shell character parameter, then no corresponding support is available in the UNIX shell environment.

The Netstat command report option

The following report options can be used with the Netstat command. If no report option is specified, Netstat displays the default CONN/**-c** report.

ALL/-A
Displays detailed information about TCP connections and UDP sockets, including some recently closed ones. See “Netstat ALL/-A report” on page 328 for more details.

ALLConn/-a
Displays information for all TCP connections and UDP sockets, including some recently closed ones. See “Netstat ALLConn/-a report” on page 362 for more details.

ARp/-R
Queries the IPv4 ARP cache information. See “Netstat ARp/-R report” on page 369 for more details.

BYTEinfo/-b
Displays the byte-count information for each active TCP connection and UDP socket. See “Netstat BYTEinfo/-b report” on page 372 for more details.

CACHinfo/-C
Displays information about TCP connections uszing the Cache Accelerator. See “Netstat CACHinfo/-C report” on page 378 for more details.

CLients/-e
Displays information about local users of TCP/IP services (jobnames). See “Netstat CLients/-e report” on page 381 for more details.

CONFIG/-f
Displays the TCP/IP configuration information about IP, TCP, UDP, SMF parameters, GLOBALCONFIG profile statement, network monitor, data trace, and autolog settings. See “Netstat CONFIG/-f report” on page 383 for more details.

COnn/-c
Displays information about each active TCP connection and UDP socket. COnn/**-c** is the default parameter. See “Netstat COnn/-c report” on page 417 for more details.

DEFADDRT/-l
Displays the policy table for IPv6 default address selection. See “Netstat DEFADDRT/-l report” on page 424 for more details.

DEvlinks/-d
Displays the information about interfaces that are defined to the TCP/IP stack. If the TCP/IP stack is using Shared Memory Communications - RDMA (SMC-R) protocols, this report option displays SMC-R links and link groups. See “Netstat DEvlinks/-d report” on page 425 for more details.

Gate/-g
Displays information about the stack routing table for IPv4 destinations. See “Netstat Gate/-g report” on page 475 for more details.

HElp or ?/-?
Displays help information for the Netstat parameters. See “Netstat HElp/-? report” on page 480 for more details.

HOme/-h
Displays information about each home IP address and its associated link or interface name. See “Netstat HOme/-h report” on page 485 for more details.

IDS/-k
Displays information about intrusion detection services. See “Netstat IDS/-k report” on page 491 for more details.

ND/-n Displays the IPv6 Neighbor cache entries. See “Netstat ND/-n report” on page 504 for more details.

PORTList/-o
Displays the list of reserved ports and the port access control configuration for unreserved ports. See “Netstat PORTList/-o report” on page 507 for more details.

RESCache/-q
Display resolver cache information. See “Netstat RESCache/-q report” on page 511 for more details.

ROUTe/-r
Displays stack routing information. Information for IPv4 destinations is always displayed. If the stack is IPv6 enabled, information about IPv6 destinations is also displayed. See “Netstat ROUTe/-r report” on page 524 for more details.

SLAP/-j
Displays the QoS policy statistics. See “Netstat SLAP/-j report” on page 539 for more details.

SOCKets/-s
Displays information for open TCP or UDP sockets associated with a client name. See “Netstat SOCKets/-s report” on page 544 for more details.

SRCIP/-J
Displays information for all job-specific and destination-specific source IP address associations on the TCP/IP address space. See “Netstat SRCIP/-J report” on page 550 for more details.

STATS/-S
Displays TCP/IP statistics for IP, ICMP, TCP, and UDP protocols. See “Netstat STATS/-S report” on page 553 for more details.

TELnet/-t
Displays information for TN3270 Telnet server connections. See "Netstat TELnet/-t report" on page 572 for more details.

TTLS/-x
Displays Application Transparent Transport Layer Security (AT-TLS) group and connection information. See "Netstat TTLS/-x report" on page 578 for more details.

Up/-u Displays the date and time that the TCP/IP stack was started and specifies whether the stack is IPv6 enabled or disabled. See "Netstat Up/-u report" on page 594 for more details.

VCRT/-V
Displays the dynamic VIPA Connection Routing Table used for sysplex distributor and moveable dynamic VIPA support. See "Netstat VCRT/-V report" on page 595 for more details.

VDPT/-O
Displays the dynamic VIPA Destination Port Table information for TCP/IP stacks, and the dynamic VIPA Destination Port Table for non-z/OS targets. See "Netstat VDPT/-O report" on page 604 for more details.

VIPADCFG/-F
Displays the dynamic VIPA configuration for a TCP/IP stack. See "Netstat VIPADCFG/-F report" on page 624 for more details.

VIPADyn/-v
Displays the current dynamic VIPA and VIPAROUTE information for a TCP/IP stack. See "Netstat VIPADyn/-v report" on page 645 for more details.

The Netstat command target

You can get information for a specific TCP/IP address space by using TCp/**-p** *tcpname* with any report option. This option is needed only if you use the Common INET physical file system (PFS) and have more than one TCP/IP address space active in a z/OS image. In such a multi-stack environment, use this option to specify which TCP/IP address space you want Netstat to gather information from. If this option is not specified in a multi-stack environment, then the information displayed is gathered only from the default TCP/IP address space that was specified with the TCPIPJOBNAME statement in the appropriate resolver configuration file or data set.

Rule: The Netstat RESCache/**-q** option gets its information from the system-wide resolver. The information is not specific to the TCP/IP address space name that is specified on the TCp/**-p** target parameter, or to the default TCP/IP address space.

TCp/-p *tcpname*
Displays detailed information about the specified TCP/IP address space. You can use TCp/**-p** *tcpname* with any other Netstat parameter to get information about the specified TCP/IP address space.

The *tcpname* is an 8-byte procedure name that is used to start the TCP/IP address space. When the **S member.identifier** method of starting TCP/IP is used, the value specified for *identifier* must be used as *tcpname*.

Netstat command output

Use the following options to specify where and in which format output is written. If an output option is not specified, by default the output is displayed on the user's terminal.

FORMat/-M

Display a Netstat report in a given format.

SHORT

Display a Netstat report in short format. The short format is the format that supports only IPv4 IP addresses. This option is valid only if the stack is not IPv6 enabled.

LONG

Display a Netstat report in long format. The long format can accommodate both IPv4 and IPv6 IP addresses.

If . . .	**Then . . .**
The stack is IPv6 enabled	The default format for the Netstat report is the long format.
The stack is IPv6 enabled and the FORMAT/**-M** SHORT is specified from the command	The error message EZZ2383I is issued and command processing is stopped.
The stack is not IPv6 enabled and the FORMAT/**-M** option is not specified from the Netstat command line nor in the IPCONFIG profile statement	The default format for Netstat report is the short format.

REPort (TSO NETSTAT only)

Causes the output to be stored in an MVS data set. If there is no additional parameter specified, the output is stored in a data set named *tsoprefix*.NETSTAT.option. If NOPREFIX is set in the TSO user profile, then the data set name is NETSTAT.*option*. The data set is created and cataloged if it does not already exist. If the data set already exists, the output from the requested option replaces any existing data. The name of the data set depends on whether either of the following additional parameters were specified:

DSN *dsnname*

Specifies the data set name in which the output is stored. The *dsnname* can be either a fully qualified name surrounded by single quotation marks (for example, 'abc.xyz') or an unqualified name (for example, abc). If an unqualified name is specified, then the unqualified name is prefixed with the TSO prefix value.

HLQ *hlqname*

Specifies the high-level qualifier for the data set in which the output is stored. The resulting data set name is *hlqname*.NETSTAT.*option*.

The following shows the relationship between the parameters and the stored data set name:

	No tsoprefix	**tsoprefix is available**
Nothing specified	NETSTAT.*option*	*tsoprefix*.NETSTAT.*option*
HLQ specified	*hlqname*.NETSTAT.*option*	*hlqname*.NETSTAT.*option*
Unqualified DSN	*dsnname*	tsoprefix.*dsnname*
Fully qualified DSN	*dsnname*	*dsnname*

Use the REPort option to store the information returned by NETSTAT in a file used for later reference. For example, to store the output of the NETSTAT COnn report in a file, issue the following command: **netstat conn report**

After you issue the preceding command, a data set named *tsoprefix*.NETSTAT.CONN is created, which contains output similar to the following information:

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
User Id  Conn     Local Socket           Foreign Socket         State
-------  ----     ------------           --------------         -----
FTPD1    0000003B 0.0.0.0..21            0.0.0.0..0             Listen
FTPD1    0000003D 9.37.65.146..21        9.67.115.5..1026       Establsh
FTPD1    0000003F 9.37.65.146..21        9.27.13.21..3711       Establsh
TCPCS    0000000F 0.0.0.0..23            0.0.0.0..0             Listen
TCPCS    0000000C 9.67.115.5..23         9.27.11.182..4886      Establsh
SYSLOGD1 00000010 0.0.0.0..514           *..*                   UDP
```

STAck (TSO NETSTAT only)

Causes the report, stripped of title lines, to be placed in the TSO data stack when NETSTAT is issued from a CLIST or a REXX EXEC. No information is displayed at the user's terminal.

TITLes

Causes the report, including title lines, to be placed in the TSO data stack when NETSTAT is issued from a CLIST or a REXX EXEC.

The Netstat command filter

The following parameters can be used to filter the output of the specified report. If you specify a filter parameter on the TSO NETSTAT command, it must be the last parameter on the command line preceded by a left parenthesis.

APPLD/-G *appldata*

Filter the output of the ALL/**-A**, ALLConn/**-a**, and COnn/**-c** reports using the specified application data *appldata*. You can enter one filter value at a time that can be 40 characters in length.

APPLname/-N *applname*

Filter the output of the TELnet/**-t** report using the specified VTAM application name *applname*. You can enter up to six filter values and each specified value can be eight characters in length.

CLIent/-E *clientname*

Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, CLient/**-e**, COnn/**-c**, SOCKets/**-s**, and TELnet/**-t** reports using the specified client name *clientname*. You can enter up to six filter values and each specified value can be eight characters in length.

CONNType/-X

Filter the report using the specified connection type. You can enter one filter value at a time.

NOTTLSPolicy

Filter the output of the ALLConn/**-a** and COnn/**-c** reports, displaying only connections that have not been matched to an Application Transparent Transport Layer Security (AT-TLS) rule. This includes connections that were established while the AT-TLS function was disabled (the value NOTTLS was specified on the TCPCONFIG statement or is in effect by default) and all connections that are not TCP protocol. For TCP connections that were established when the AT-TLS function was enabled, this includes the following connections:

- Connections for which AT-TLS policy lookup has not yet occurred (typically the first send or receive has not yet been issued)
- Connections for which AT-TLS policy lookup has occurred but no matching rule was found

TTLSPolicy

Filter the output of the ALLConn/**-a** and COnn/**-c** reports, displaying only connections that match an Application Transparent Transport Layer Security (AT-TLS) rule. This includes only connections that were established while the AT-TLS function was enabled, for which an AT-TLS policy rule was found with the value `TTLSEnabled ON` or `TTLSEnabled OFF` specified in the TTLSGroupAction. Responses can be further limited on AT-TLS connection type. AT-TLS connection type has the following values:

CURRent

Display only connections that are using AT-TLS where the rule and all actions are still available to be used for new connections.

GRoup *groupid*

Display only connections that are using the AT-TLSgroup specified by the *groupid* value. The specified *groupid* value is a number that is assigned by the TCP/IP stack to uniquely identify an AT-TLS group. You can determine the *groupid* value from the GroupID field that is displayed in the Netstat TTLS/**-x** GROUP report.

STALE

Display only connections that are using AT-TLS where the rule or at least one action is no longer available to be used for new connections.

DNSAddr/-Q *dnsaddr*

Filter the output of the RESCache/**-q** report using the specified DNS IP address *dnsaddr*. You can enter one filter value at a time. The specified IPv4 *dnsaddr* value can be 1–15 characters in length; the specified IPv6 *dnsaddr* value can be 1–45 characters in length.

Restriction: The DNSAddr/**-Q** filter does not support wildcard characters.

HOSTName/-H *hostname*

Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, COnn/**-c**, RESCache/**-q**, SOCKets/**-s**, TELnet/**-t**, and VCRT/**-V** reports using the specified host name value *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters in length.

Result: For reports other than those produced using the RESCache/**-q** option, at the end of the report, the Netstat command displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter supports wildcard characters only for the RESCache/**-q** option, but not for other options.

2. With options other than the RESCache/**-q** option, using the HOSTName filter might cause delays in the output because the *hostname* value must be resolved (depending on resolver and DNS configuration).
3. For the RESCache/**-q** option, the HOSTName/**-H** filter applies only to the HostName to IPAddress translation portion of the report.

INTFName/-K *intfname*

Filter the output of the DEvlinks/**-d** and HOme/**-h** reports using the specified interface name value *intfname*. You can enter one filter value at a time and the specified value can be 1–16 characters in length.

For the DEvlinks and HOme report options, the INTFName filter can be one of the following names:

- The link name of a network interface that was configured on a LINK profile statement (this option selects one interface).
- The interface name of a network interface that was configured on an INTERFACE profile statement (this option selects one interface).
- The port name of an OSA-Express feature in QDIO mode. This is the name that is specified on the PORTNAME keyword in the TRLE (this option selects all interfaces that are associated with the OSA-Express port, including an OSAENTA trace interface).
- The name of a HiperSockets TRLE (this option selects all interfaces that are associated with the HiperSockets TRLE).

Additionally, for the DEvlinks report option, the INTFName filter can also be the interface name of an OSAENTA trace interface, which is EZANTA*portname*, where the *portname* value is the name that is specified on the PORTNAME keyword in the TRLE for the OSA-Express port that is being traced (this option selects one interface).

Guideline: For the DEvlinks/**-d** option, if a network resource has been coded in TCPIP.PROFILE using the DEVICE/LINK/HOME statements, then the *intfname* value that should be used is the link name that was specified on the LINK profile statement. Otherwise, use the interface name that was specified on the INTERFACE profile statement.

Restriction: The INTFName filter does not support wildcard characters.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. For options other than the RESCache/**-q** option, you can enter up to six filter values; the RECache**/-q** option accepts only one filter value at a time in *ipaddr* format. Each specified IPv4 *ipaddr* value can be 1–15 characters in length and each selected IPv6 *ipaddr* value can be 1–45 characters in length.

ipaddr Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, COnn/**-c**, Gate/**-g**, ND/**-n**, RESCache/**-q**, ROUTe/**-r**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, VDPT/**-O**, and VIPADCFG/**-F** reports using the specified IP address *ipaddr*. For all options except the RESCache/**-q** option, the default subnet mask 255.255.255.255 is used for IPv4 addresses; for IPv6 addresses, the default *prefixlength* value 128 is used. The RECache/**-q** option does not support any default subnet mask or default *prefixlength* values.

ipaddr/prefixlength

Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, COnn/**-c**, ND/**-n**, ROUTe/**-r**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, VDPT/**-O**, and VIPADCFG/**-F** reports using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, COnn/**-c**, Gate/**-g**, ROUTe/**-r**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, VDPT/**-O**, and VIPADCFG/**-F** reports using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Note:

1. For the Gate/**-g** option, *ipaddr* is the destination IP address; it is not the destination network address.
2. When filtering Gate/**-g** and ROUTe/**-r** outputs on a specified IP address, the DEFAULT and DEFAULTNET routes are not displayed.

Guidelines:

- For ALL/**-A**, ALLConn/**-a**, COnn/**-c**, and TELnet/**-t** options, *ipaddr* can be either the local or remote IP address. For the BYTEinfo/**-b** option, *ipaddr* can be a remote IP address. For the SOCKets/**-s** option, *ipaddr* can be an address to which the socket is bound or connected. For the VCRT/**-V** option, *ipaddr* can be a source IP address, a destination IP address, or a destination XCF IP address. For the VDPT/**-O** option, *ipaddr* can be a destination IP address or a destination XCF IP address. For the VIPADCFG/**-F** option, *ipaddr* can be a dynamic VIPA address, a destination IP address, or a destination XCF IP address.
- For an IPv6-enabled stack (except for RESCache/**-q** option):
  - Both IPv4 and IPv6, *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
  - For an IPv6-enabled stack, an IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as its IPv4 address. But, for ROUTE/**-r** and ND/**-n** options, an IPv4-mapped IPv6 address is treated as an IPv6 address. If an IPv4-mapped IPv6 address is entered as an *ipaddr* value for these two options, no matching entry is found.
- For the RESCache/**-q** option, the *ipaddr* value can be either an IPv4 or IPv6 address regardless of whether the stack is configured for IPv4 or IPv6 operation.

Restrictions:

- The IPAddr/**-I** filter for RESCache/**-q**, VCRT/**-V**, VDPT/**-O**, and VIPADCFG/**-F** options does not support wildcard characters.
- The IPAddr/**-I** filter for an IPv6 address does not support wildcard characters.
- For a UDP endpoint socket, the filter value applies only to the local or source IP address.
- For all options except the RESCache/**-q** option, for an IPv4-only stack, only IPv4 *ipaddr* values are accepted. The RECache/**-q** option always accepts IPv4 and IPv6 addresses, regardless of the capability of the stack.

- For the ND/**-n** option, an IPv4 *ipaddr* value is not accepted.
- For the RESCache/**-q** option, the IPAddr/-I filter applies only to the IPAddress to HostName translation portion of the report.
- The RECache/**-q** option accepts only one filter value at a time in *ipaddr* format.

IPPort/-B *ipaddr+portnum*
Filter the report output of the ALL**/-A**, ALLConn/**-a**, CONN/**-c**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, and VDPT/**-O** reports using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- For the ALL/**-A**, ALLConn/**-a**, COnn/**-c**, and TELnet/**-t** options, the *ipaddr* value can be either the local or remote IP address. For the SOCKets/**-s** option, the *ipaddr* value can be an address to which the socket is bound or connected. For the VCRT/**-V** option, the *ipaddr* value can be a source IP address, a destination IP address, or a destination XCF IP address. For the VDPT/**-O** option, the *ipaddr* value can be a destination IP address or a destination XCF IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.
- An entry is returned only when both the *ipaddr* and *portnum* values match.

LUName/-L *luname*
Filter the output of the TELnet/**-t** report using the specified LU name *luname*. You can enter up to six filter values and each specified value can be up to eight characters in length.

NOTN3270/-T
Filter the output of the ALL/**-A**, ALLConn/**-A**, BYTEinfo/**-b**, CLient/**-e**, COnn/**-c**, and SOCKets/**-s** reports, excluding TN3270 server connections.

POLicyn/-Y *policyname*
Filter the output of the SLAP/**-j** report using the specified policy rule name *policyname*. You can enter one filter value at a time and the specified value can be up to 48 characters in length.

POrt/-P *portnum*
Filter the output of the ALL/**-A**, ALLConn/**-a**, COnn/**-c**, PORTList/**-o**,

SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, and VDPT/**-O** reports using the specified port number *portnum*. You can enter up to six filter values.

Guidelines:

- The port number can be either a local port or a remote port.

  For the SOCKets/**-s** option, the port can be a port to which the socket is bound or connected.
- For the ALL/**-A**, ALLConn/**-a**, COnn/**-c**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, and VDPT/**-O** reports, the port value range is 0-65535
- For the PORTList/**-o** option only, the port value range is 1-65535 and you can also filter on the keyword UNRSV

Restriction:

- No wildcards are allowed.
- For a UDP endpoint socket, the filter value applies only to the local or source IP address.

SMCID/-U *smcid*

Filter the output of the ALL/-A, ALLConn/-a, COnn/-c, and DEvlinks/-d reports by using the specified Shared Memory Communications over Remote Direct Memory Access (SMC-R) link or link group identifier *smcid*. If an asterisk (*) is specified for the filter value, Netstat provides output only for the entries that are associated with SMC-R link, and link groups. You can enter one filter value at a time.

Except for POrt/**-P**, INTFName/**-K**, CONNType/**-X** TTLSPolicy GRoup *groupid*, DNSAddr/**-Q**, SMCID/**-U** and IPPort/**-B**, the filter value can be a complete or partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string *searchee* matches with **ar?he**, but the string *searhee* does not match with **ar?he**. If you want to use the wildcard character on the IPAddr/**-I** parameter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/subnetmask* or *ipaddr/prefixlen* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care when you use a z/OS UNIX MVS special character in a character string such as using a wildcard character in a filter value. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -g -I '10.*.0.0'** or **netstat -g -I "10.*.0.0"**.

Command to terminate a socket endpoint

You can terminate a specific TCP/IP socket end-point using the following command:

DRop/-D *n*

Terminates the socket endpoint that is identified by the connection number *n*. You can determine the connection number from the *Conn* column in the Netstat COnn/**-c** or Netstat TELnet/**-t** display. You can use this parameter only if the OPERCMDS class resource MVS.VARY.TCPIP.DROP is defined to the security product (such as RACF) and the user ID associated with the

DRop/**-D** command is permitted to this resource. See “Netstat DRop/-D command” on page 474 for detailed information.

Netstat report details and examples

In order to fully understand the following concepts and fields, you need to have some general knowledge of TCP/IP. See the IBM Redbooks publication *TCP/IP tutorial and Technical Overview*, GG24-3376 for more information.

Netstat report general concept

TCP connection status:
A TCP connection progresses through a series of states during its lifetime. The following diagram illustrates the possible states for a TCP connection and how the states transition based on various events from either the network or from the local TCP sockets application.

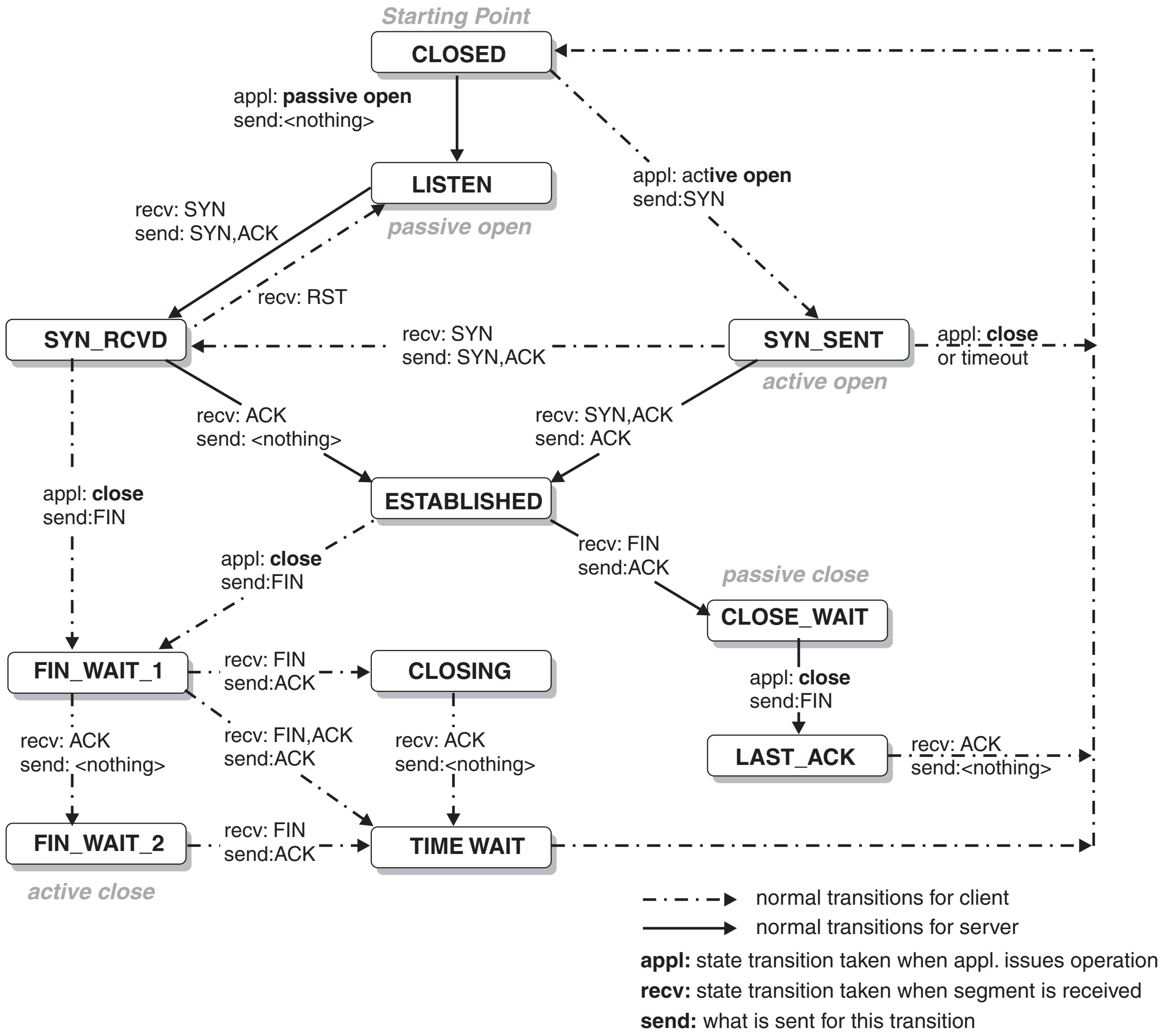


Figure 1. TCP state transition diagram

Table 15. TCP state transition description table

TCP connection state	Abbreviation in MVS console	Abbreviation in TSO or UNIX shell	Description
LISTEN	Listen	Listen	Waiting for a connection request from a remote TCP application. This is the state in which you can find the listening socket of a local TCP server.
SYN-SENT	SynSent	SynSent	Waiting for an acknowledgment from the remote endpoint after having sent a connection request. Results after step 1 of the three-way TCP handshake.
SYN-RECEIVED	SynRcvd	SynRcvd	This endpoint has received a connection request and sent an acknowledgment. This endpoint is waiting for final acknowledgment that the other endpoint did receive this endpoint's acknowledgment of the original connection request. Results after step 2 of the three-way TCP handshake.
ESTABLISHED	Estblsh	Establsh	Represents a fully established connection; this is the normal state for the data transfer phase of the connection.
FIN-WAIT-1	FinWt1	FinWait1	Waiting for an acknowledgment of the connection termination request or for a simultaneous connection termination request from the remote TCP. This state is normally of short duration.
FIN-WAIT-2	FinWt2	FinWait2	Waiting for a connection termination request from the remote TCP after this endpoint has sent its connection termination request. This state is normally of short duration, but if the remote socket endpoint does not close its socket shortly after it has received information that this socket endpoint closed the connection, then it might last for some time. Excessive FIN-WAIT-2 states can indicate an error in the coding of the remote application.
CLOSE-WAIT	ClosWt	ClosWait	This endpoint has received a close request from the remote endpoint and this TCP is now waiting for a connection termination request from the local application.
CLOSING	Closing	Closing	Waiting for a connection termination request acknowledgment from the remote TCP. This state is entered when this endpoint receives a close request from the local application, sends a termination request to the remote endpoint, and receives a termination request before it receives the acknowledgment from the remote endpoint.
LAST-ACK	LastAck	LastAck	Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP. This state is entered when this endpoint received a termination request before it sent its termination request.

Table 15. TCP state transition description table (continued)

TCP connection state	Abbreviation in MVS console	Abbreviation in TSO or UNIX shell	Description
TIME-WAIT	TimeWt	TimeWait	Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
CLOSED	Closed	Closed	Represents no connection state at all.

Clients or Users

For various reasons, TCP/IP refers to MVS jobs or address spaces that use TCP/IP services as clients or users of TCP/IP services. The term client in this context has nothing to do with the traditional client/server roles of a network application. Both local server programs and local client programs on z/OS are clients or users of TCP/IP services. For most purposes you can substitute Client name, User ID, and User in the Netstat reports with MVS *jobname*.

UDP socket status

UDP, unlike TCP, does not operate with strict states. The state that is shown in the various Netstat reports is always UDP for UDP sockets.

Client ID or Connection number

A generated number that uniquely identifies a socket endpoint that might represent a connection on this TCP/IP host. This number can be used to drop a socket or connection with the Netstat DROP/**-D** parameter.

Client name or User ID

The client name from a TCP/IP perspective is in general the job name of the address space that owns the socket. For batch jobs this is the job name. For TSO users, this is the TSO user ID. For UNIX processes this is the job name as determined during process creation, either by appending a digit to the job name (INETD creates INETD1) of the parent process or by setting the job name to the value of the _BPX_JOBNAME environment variable. For started tasks, the job name is generally the procedure name. If a procedure is started with the JOBNAME keyword (S procname,JOBNAME=myjob), then the job name becomes the value that was specified on that JOBNAME keyword. If a procedure is started with a start modifier (S procname.modif), then the modifier is what is shown as the TCP/IP client name.

Local IP address

A socket might have no address information at all (right after it has been created by a program using the socket() call); it might have just a local address (a local IP address and/or a local port number) that was set using a bind() socket call; or it might have both a local address and a remote address, in which case it represents a connected socket (a socket that is in connection with a remote socket).

The local IP address of a socket is either zero (not bound to any local IP address) or it is an IP address that is in the HOME list of this TCP/IP host.

The listening socket of a server program has only the local address filled in. If the local IP address of the server's listening socket is zero, then remote clients are allowed to send connection requests to any IP address that is in this TCP/IP host's HOME list. If the local IP address of the server's listening socket is nonzero, then remote clients can connect to this server only by sending connection requests to that specific IP address. A connected socket has both the local and the remote address filled in.

Foreign/remote IP address

The remote IP address is present for connected sockets and represents the IP address that is associated with the remote socket endpoint to which this socket is connected. A connected socket might be one of the following sockets:

- A server socket where the remote client that is represented by this remote IP address connected to a server on this TCP/IP host.
- A socket belonging to a client program on this TCP/IP host that is connected to a server on the remote TCP/IP host that is represented by this remote IP address.

Local port

The local port is part of the local address of a socket. For a server's listening socket, the port represents the specific server. If remote clients need to use the services of this server, they send a connection request to this TCP/IP host to this server's specific port number.

Connected sockets might represent one of the following case:

- A connection with a local server from a remote client, for example, the local port number is the same port number that appears on the server's listening socket.
- A local client connected to a remote server, for example, the port number could be any port number the TCP/IP host found available when the connection was being established (also known as an ephemeral or short-lived port number). This is typically a port number higher than 1024.

Foreign/remote port

The remote port is part of the remote address of a socket and is present only for connected sockets. It represents the port number of the remote socket that is connected to this socket. If the connected socket belongs to a client program on this TCP/IP host, then the remote port number identifies the server on the remote TCP/IP host to which this client program is connected.

Local socket

The IP address and port number to which the application on the local stack was bound.

Foreign socket

The IP address and port number to which the application on the remote host was bound. For UDP sockets, the foreign socket field that is shown in the various Netstat reports is displayed as *..* if the socket is not connected. For connected UDP sockets, the foreign socket field shows the remote IP address and port specified on the connect request. When a UDP socket is connected, it accepts packets only from the specified remote IP address and port.

Last touched time

For TCP, the last time one of the following events occurred to the connection:

- The server side receives a connection request.
- The server side accepts the connection request.
- Either the server or client side of a connection receives a packet.
- Either the server or client side of a connection sends a packet.

For UDP, the last time one of the following events occurred to the connection:

- Either the server or client side of a connection receives a packet.
- Either the server or client side of a connection sends a packet.

Redirecting Netstat output:

Netstat screen output can be redirected for all Netstat reports. The following example uses the BYTEINFO report:

From TSO environment:

- You can redirect TSO NETSTAT screen output to a disk file by appending a REPORT option.

  NETSTAT BYTEINFO REPORT

  The data set MVSUSER.NETSTAT.BYTEINFO (where MVSUSER is the user ID) is created containing the screen output from a BYTEINFO command. See “Netstat command output” on page 316 for more description of the REPORT option.

- You can also redirect TSO NETSTAT screen output to the TSO data stack by appending a STACK option.

  NETSTAT BYTEINFO STACK

  Causes the report, stripped of title lines, to be placed in the TSO data stack containing the screen output from a BYTEINFO command. See “Netstat command output” on page 316 for more description of the STACK option.

From z/OS UNIX shell environment:

You can redirect the netstat screen output to a file by using the redirect function (>) in the following format:

```
netstat -b > byteinfo
```

The file byteinfo is created in your current directory containing the screen output shown previously.

Time stamp

The time stamp displayed in the header for each Netstat report is in local time. The time field displayed in reports ALL/**-A**, BYTEinfo/**-b**, CLients/**-e**, HOME/**-h**, RESCache/**-q**, ROUTe/**-r**, SLAP/**-j**, UP/**-u**, and VIPADyn/**-v** is Coordinated Universal Time (UTC). UTC time does not take leap seconds into account.

Netstat ALL/-A report

Displays detailed information about TCP connections and UDP sockets, including some recently closed ones. The purpose of this report is to aid in debugging problems with TCP connections and UDP sockets.

TSO syntax:

```
►►──NETSTAT  ALL──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────►◄
```

Modifier:

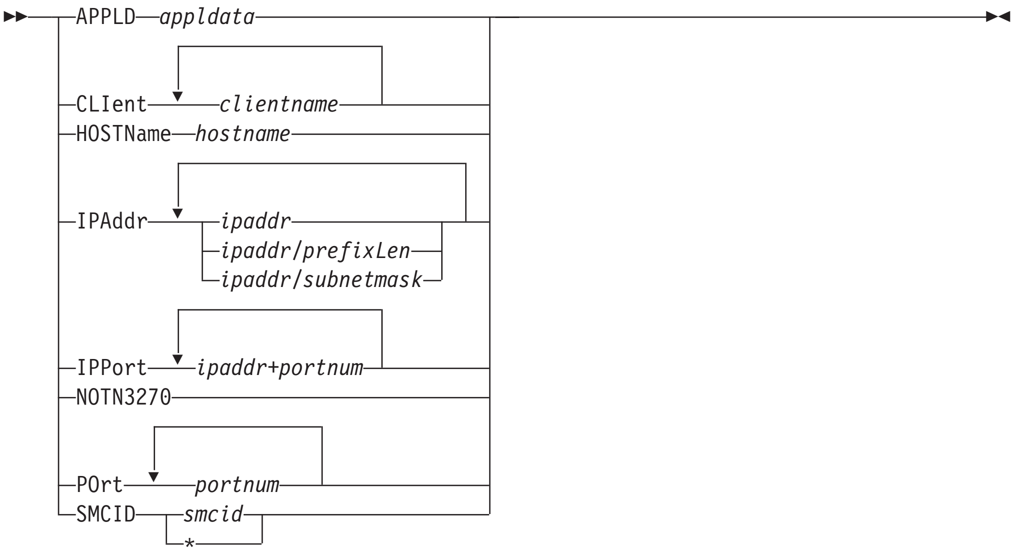
```
►►──┤ SERVER ├──────────────────────────────────────────────────────────────►◄
```

SERVER
: Provide detailed information only for TCP connections that are in the listen state.

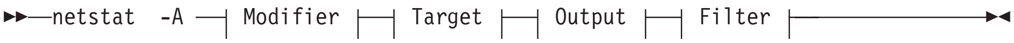
Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

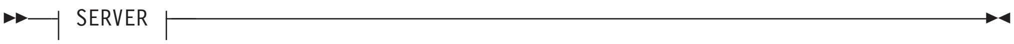
Filter:



Netstat ALL/-A report z/OS UNIX syntax:



Modifier:



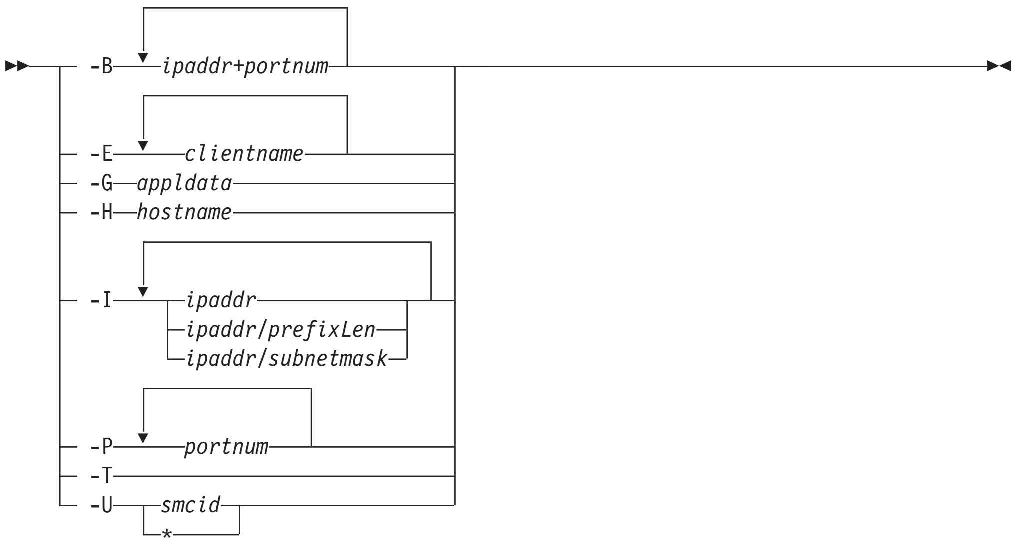
SERVER
: Provide detailed information only for TCP connections that are in the listen state.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See “The Netstat command target” on page 316 for more information about the **-p** parameter.

Output:

The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:



Filter description:

APPLD/-G *appldata*
: Filter the output of the ALL/**-A** report using the specified application data *appldata*. You can enter one filter value at a time and the specified value can be 40 characters in length.

CLIent/-E *clientname*
: Filter the output of the ALL/**-A** report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be eight characters in length.

HOSTName/-H *hostname*
: Filter the output of the ALL/**-A** report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters in length.

Result: At the end of the report, Netstat displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using the HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*
: Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be 45 characters in length.

ipaddr Filter the output of the ALL/**-A** report using the specified IP address *ipaddr*. For IPv6 addresses, the default *prefixlength* 128 is used.

ipaddr/prefixlength
Filter the output of the ALL/**-A** report using a specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask
Filter the output of the ALL/**-A** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be either the local or remote IP address.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as its IPv4 address.

Restrictions:

1. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
2. The filter value for an IPv6 address does not support wildcard characters.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

IPPort/-B *ipaddr***+***portnum*
Filter the report output of the ALL/**-A** report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be either the local or remote IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.

- For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.

NOTN3270/-T
Filter the output of the ALL/**-A** report, excluding TN3270 server connections.

POrt/-P *portnum*
Filter the output of the ALL/**-A** report using the specified port number *portnum*. You can enter up to six filter values. For all *portnum* values that were reserved by the same PORTRANGE profile statement, only one output line is displayed.

Guideline: The port number can be either a local or remote port.

Restriction: For a UDP endpoint socket, the filter value applies only to the local or source port.

SMCID/-U *smcid*
Filter the output of the ALL/-A report by using the specified Shared Memory Communications over Remote Direct Memory Access (SMC-R) link or link group identifier *smcid*. If an asterisk (*) is specified for the filter value, Netstat provides output only for the entries that are associated with SMC-R link, and link groups. You can enter one filter value at a time.

The filter value for CLIent/**-E**, IPAddr/**-I**, and APPLD/-G can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string *searchee* matches with **ar?he**, but the string *searhee* does not match **ar?he**. To use the wildcard character on the IPAddr/**-I** filter, specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/prefixlen* or *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -A -I '10.*.0.0'**.

Command syntax examples:

From TSO environment:

```
NETSTAT ALL
   Display detailed information about TCP connections and UDP sockets in the default
   TCP/IP stack.
NETSTAT ALL TCP TCPCS6
   Display detailed information about TCP connections and UDP sockets in TCPCS6 stack.
NETSTAT ALL TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display detailed information about those TCP connections and UDP sockets in TCPCS8
   stack whose local or remote IP addresses match the specified filter IP address values.
NETSTAT ALL (PORT 2222 6666 88
   Display detailed information about those TCP connections and UDP sockets in the
   default TCP/IP stack whose local or remote ports match the specified filter port
   numbers.
NETSTAT ALL SERVER TCP TCPCS
   Display detailed information about those TCP connections in listen state on
   TCP/IP stack TCPCS
NETSTAT ALL TCP TCPCS (IPPORT 127.0.0.1+21
   Display detailed information about connections using ip address 127.0.0.1 and
   port 21 on TCP/IP stack TCPCS
```

From UNIX shell environment:

```
   netstat -A
   netstat -A -p tcpcs6
   netstat -A -p tcpcs6 -I 9.43.1.1 9.43.2.2
   netstat -A -P 2222 6666 88
   netstat -A SERVER -p tcpcs
   netstat -A -B 127.0.0.1+21 -p tcpcs
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT ALL
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           22:24:30
Client Name: FTPD1                    Client Id: 000000F9
Local Socket: 9.42.104.43..21         Foreign Socket: 9.42.103.165..1035
  BytesIn:            0000000035        BytesOut:           0000000265
  SegmentsIn:         0000000017        SegmentsOut:        0000000014
  StartDate:          01/09/2012        StartTime:          22:04:11
  Last Touched:       22:04:18          State:              Establsh
  RcvNxt:             0214444666        SndNxt:             0216505563
  ClientRcvNxt:       0214443596        ClientSndNxt:       0216504670
  InitRcvSeqNum:      0214443560        InitSndSeqNum:      0216504404
  CongestionWindow:   0000007336        SlowStartThreshold: 0000065535
  IncomingWindowNum:  0214477396        OutgoingWindowNum:  0216538247
  SndWl1:             0214444666        SndWl2:             0216505563
  SndWnd:             0000032684        MaxSndWnd:          0000032768
  SndUna:             0216505563        rtt_seq:            0216505479
  MaximumSegmentSize: 0000000524        DSField:            00
  Round-trip information:
    Smooth trip time: 102.000           SmoothTripVariance: 286.000
  ReXmt:              0000000000        ReXmtCount:         0000000000
  DupACKs:            0000000000        RcvWnd:             0000032730
  SockOpt:            85                TcpTimer:           00
  TcpSig:             84                TcpSel:             60
  TcpDet:             E0                TcpPol:             00
  TcpPrf:             40
  QOSPolicy:          No
  TTLSPolicy:         Yes
    TTLSRule:         server
    TTLSGrpAction:    group_action1
    TTLSEnvAction:    Environment1
  RoutingPolicy:      No
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ReceiveDataQueued:  0000000000
  SendDataQueued:     0000000000
  SendStalled:        No
  Ancillary Input Queue: Yes
    BulkDataIntfName: OSAQDIO4
  Application Data:   EZAFTP0S C USER1     PTS305
----
```

```
Client Name: FTPD1                    Client Id: 000000F6
Local Socket: 0.0.0.0..21             Foreign Socket: 0.0.0.0..0
  BytesIn:            0000000000        BytesOut:           0000000000
  SegmentsIn:         0000000000        SegmentsOut:        0000000000
  StartDate:          01/09/2012        StartTime:          22:04:11
  Last Touched:       21:41:09          State:              Listen
  RcvNxt:             0000000000        SndNxt:             0000000000
  ClientRcvNxt:       0000000000        ClientSndNxt:       0000000000
  InitRcvSeqNum:      0000000000        InitSndSeqNum:      0000000000
  CongestionWindow:   0000000000        SlowStartThreshold: 0000000000
  IncomingWindowNum:  0000000000        OutgoingWindowNum:  0000000000
  SndWl1:             0000000000        SndWl2:             0000000000
  SndWnd:             0000000000        MaxSndWnd:          0000000000
  SndUna:             0000000000        rtt_seq:            0000000000
  MaximumSegmentSize: 0000000536        DSField:            00
  Round-trip information:
    Smooth trip time: 0.000             SmoothTripVariance: 1500.000
  ReXmt:              0000000000        ReXmtCount:         0000000000
  DupACKs:            0000000000        RcvWnd:             0000032768
  SockOpt:            80                TcpTimer:           00
  TcpSig:             00                TcpSel:             20
  TcpDet:             C0                TcpPol:             00
  TcpPrf:             40
  QOSPolicy:          No
  TTLSPolicy:         No
  RoutingPolicy:      No
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ConnectionsIn:      0000000001        ConnectionsDropped: 0000000000
  MaximumBacklog:     0000000010        ConnectionFlood:    No
  CurrentBacklog:     0000000000
    ServerBacklog:    0000000000        FRCABacklog:        0000000000
  CurrentConnections: 0000000001        SEF:                98
  Quiesced:           No
  SharePort: WLM
    RawWeight:        63                NormalizedWeight:   15
    Abnorm:           10                Health:             100
    RawCP:  060       RawzAAP:  000     RawzIIP:  040
    PropCP: 040       PropzAAP: 000     PropzIIP: 023
    ILWeighting: 1    XcostzAAP: 001    XcostzIIP: 001
  Application Data:   EZAFTP0D
----
```

```
Client Name: TCPCS                    Client Id: 0000000C
Local Socket: 9.67.115.5..23          Foreign Socket: 9.27.11.182..4665
  BytesIn:            0000001062        BytesOut:           0000000480
  SegmentsIn:         0000000019        SegmentsOut:        0000000019
  StartDate:          01/09/2012        StartTime:          16:46:15
  Last Touched:       16:46:15          State:              Establsh
  RcvNxt:             3296375906        SndNxt:             3296308452
  ClientRcvNxt:       3296375906        ClientSndNxt:       3296308452
  InitRcvSeqNum:      3296374843        InitSndSeqNum:      3296307971
  CongestionWindow:   0000340353        SlowStartThreshold: 0000016384
  IncomingWindowNum:  3296408638        OutgoingWindowNum:  3296341180
  SndWl1:             3296375906        SndWl2:             3296308452
  SndWnd:             0000032728        MaxSndWnd:          0000032768
  SndUna:             3296308452        rtt_seq:            3296308412
  MaximumSegmentSize: 0000065483        DSField:            00
  Round-trip information:
    Smooth trip time: 37.000            SmoothTripVariance: 101.000
  ReXmt:              0000000000        ReXmtCount:         0000000000
  DupACKs:            0000000000
  SockOpt:            00                TcpTimer:           00
  TcpSig:             00                TcpSel:             C0
  TcpDet:             F0                TcpPol:             00
  TcpPrf:             40
  QOSPolicy:          Yes
    QOSRuleName:      QosRule1
  TTLSPolicy:         Yes
    TTLSRule:         TTLSRule1
    TTLSGrpAction:    TTLSGrpAction1
    TTLSEnvAction:    TTLSEnvAction1
    TTLSConnAction:   TTLSConnAction1 (Stale)
  RoutingPolicy:      Yes
    RoutingTableName: prTabl
    RoutingRuleName:  SecLow2
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ReceiveDataQueued:  000000002C
    OldQDate:         09/15/06          OldQTime:           03:36:32
  SendDataQueued:     000002C000
    OldQDate:         09/15/06          OldQTime:           03:36:32
  SendStalled:        No
  SMC information:
    SMCStatus:        Active            SMCGroupId:         00000100
    LocalSMCLinkId:   00000101          RemoteSMCLinkId:    00000301
    LocalSMCRcvBuf:   64K               RemoteSMCRcvBuf:    64K
  Ancillary Input Queue: N/A
  Application Data:   EZBTNSRV TCPM1001 TSO10002 ET ST14S
----
Client Name: APPV4                    Client Id: 00000015
Local Socket: 0.0.0.0..2049           Foreign Socket: 9.42.103.99..1234
  BytesIn:            0000000200        BytesOut:           0000000100
  DgramIn:            0000000010        DgramOut:           0000000005
  StartDate:          06/16/2011        StartTime:          22:53:55
  Last Touched:       16:00:29
  MaxSendLim:         0000065535        MaxRecvLim:         0000065535
  SockOpt:            00                DSField:            00
  QOSPolicy:          Yes
    QOSRuleName:      QosRule2
  RoutingPolicy:      Yes
    RoutingTableName: prTab4
    RoutingRuleName:  SecLow4
  ReceiveDataQueued:  0000000000        ReceiveMsgCnt:      0000000000
  ----
Client Name: SYSLOGD1                 Client Id: 00000010
Local Socket: 0.0.0.0..514            Foreign Socket: *..*
  BytesIn:            0000000000        BytesOut:           0000000000
  DgramIn:            0000000000        DgramOut:           0000000000
  StartDate:          06/16/2011        StartTime:          23:33:52
  Last Touched:       16:46:29
  MaxSendLim:         0000065535        MaxRecvLim:         0000065535
  SockOpt:            00                DSField:            00
  QOSPolicy:          No
  RoutingPolicy:      No
  ReceiveDataQueued:  0000000000        ReceiveMsgCnt:      0000000000
----
```

IPv6 enabled or request for LONG format:

```
NETSTAT ALL
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           22:06:44
Client Name: FTPD1                    Client Id: 0000006D
  Local Socket: ::1..21
  Foreign Socket: ::1..1026
    BytesIn:            00000000000000000035
    BytesOut:           00000000000000000265
    SegmentsIn:         00000000000000000015
    SegmentsOut:        00000000000000000015
    StartDate:          01/09/2012       StartTime:          22:04:11
    Last Touched:       22:05:51         State:              Establsh
    RcvNxt:             0634886921       SndNxt:             0634950319
    ClientRcvNxt:       0634885851       ClientSndNxt:       0634949426
    InitRcvSeqNum:      0634885815       InitSndSeqNum:      0634949160
    CongestionWindow:   0000299155       SlowStartThreshold: 0000065535
    IncomingWindowNum:  0634919651       OutgoingWindowNum:  0634983003
    SndWl1:             0634886921       SndWl2:             0634950319
    SndWnd:             0000032684       MaxSndWnd:          0000032768
    SndUna:             0634950319       rtt_seq:            0634950235
    MaximumSegmentSize: 0000065463       DSField:            00
    Round-trip information:
      Smooth trip time: 81.000           SmoothTripVariance: 270.000
    ReXmt:              0000000000       ReXmtCount:         0000000000
    DupACKs:            0000000000       RcvWnd:             0000032730
    SockOpt:            8500             TcpTimer:           00
    TcpSig:             85               TcpSel:             64
    TcpDet:             E0               TcpPol:             00
    TcpPrf:             40
    QOSPolicy:          No
    TTLSPolicy:         Yes
      TTLSRule:         server
      TTLSGrpAction:    group_action1
      TTLSEnvAction:    Environment1
    RoutingPolicy:      No
    ReceiveBufferSize:  0000016384       SendBufferSize:     0000016384
    ReceiveDataQueued:  0000000000
    SendDataQueued:     0000000000
    SendStalled:        No
    Ancillary Input Queue: Yes
      BulkDataIntfName: OSAQDIO4
    Application Data:   EZAFTP0S C USER1     PTS305
----
```

```
Client Name: FTPD1                    Client Id: 0000005B
  Local Socket: ::..21
  Foreign Socket: ::..0
    BytesIn:            00000000000000000000
    BytesOut:           00000000000000000000
    SegmentsIn:         00000000000000000000
    SegmentsOut:        00000000000000000000
    StartDate:          01/09/2012       StartTime:          22:05:11
    Last Touched:       22:05:41         State:              Listen
    RcvNxt:             0000000000       SndNxt:             0000000000
    ClientRcvNxt:       0000000000       ClientSndNxt:       0000000000
    InitRcvSeqNum:      0000000000       InitSndSeqNum:      0000000000
    CongestionWindow:   0000000000       SlowStartThreshold: 0000000000
    IncomingWindowNum:  0000000000       OutgoingWindowNum:  0000000000
    SndWl1:             0000000000       SndWl2:             0000000000
    SndWnd:             0000000000       MaxSndWnd:          0000000000
    SndUna:             0000000000       rtt_seq:            0000000000
    MaximumSegmentSize: 0000000536       DSField:            00
    Round-trip information:
      Smooth trip time: 0.000            SmoothTripVariance: 1500.000
    ReXmt:              0000000000       ReXmtCount:         0000000000
    DupACKs:            0000000000       RcvWnd:             0000032768
    SockOpt:            8000             TcpTimer:           00
    TcpSig:             01               TcpSel:             20
    TcpDet:             C0               TcpPol:             00
    TcpPrf:             40
    QOSPolicy:          No
    TTLSPolicy:         No
    RoutingPolicy:      No
    ReceiveBufferSize:  0000016384       SendBufferSize:     0000016384
    ConnectionsIn:      0000000001       ConnectionsDropped: 0000000000
    MaximumBacklog:     0000000010       ConnectionFlood:    No
    CurrentBacklog:     0000000000
      ServerBacklog:    0000000000       FRCABacklog:        0000000000
    CurrentConnections: 0000000001       SEF:                100
    Quiesced:           No
    SharePort: WLM
      RawWeight:        63               NormalizedWeight:   15
      Abnorm:           10               Health:             100
      RawCP:  060       RawzAAP:  000    RawzIIP:  040
      PropCP: 040       PropzAAP: 000    PropzIIP: 023
    Application Data:   EZAFTP0D
----
```

```
Client Name: TCPCS                    Client Id: 0000001E
  Local Socket: 9.67.115.5..23
  Foreign Socket: 9.27.11.182..4665
    BytesIn:            00000000000000001062
    BytesOut:           00000000000000000480
    SegmentsIn:         00000000000000000019
    SegmentsOut:        00000000000000000018
    StartDate:          01/09/2012        StartTime:          14:27:37
    Last Touched:       14:27:37          State:              Establsh
    RcvNxt:             2776729719        SndNxt:             2776682484
    ClientRcvNxt:       2776729719        ClientSndNxt:       2776682484
    InitRcvSeqNum:      2776728656        InitSndSeqNum:      2776682003
    CongestionWindow:   0000340353        SlowStartThreshold: 0000016384
    IncomingWindowNum:  2776762451        OutgoingWindowNum:  2776715212
    SndWl1:             2776729719        SndWl2:             2776682484
    SndWnd:             0000032728        MaxSndWnd:          0000032768
    SndUna:             2776682484        rtt_seq:            2776682444
    MaximumSegmentSize: 0000065483        DSField:            00
    Round-trip information:
      Smooth trip time: 100.000           SmoothTripVariance: 163.000
    ReXmt:              0000000000        ReXmtCount:         0000000000
    DupACKs:            0000000000
    SockOpt:            0000              TcpTimer:           00
    TcpSig:             00                TcpSel:             C0
    TcpDet:             F0                TcpPol:             00
    TcpPrf:             40
    QOSPolicy:          Yes
      QOSRuleName:      QosRule1
    TTLSPolicy:         Yes
      TTLSRule:         TTLSRule1
      TTLSGrpAction:    TTLSGrpAction1
      TTLSEnvAction:    TTLSEnvAction1
      TTLSConnAction:   TTLSConnAction1 (Stale)
    RoutingPolicy:      Yes
      RoutingTableName: prTabl
      RoutingRuleName:  SecLow2
    ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
    ReceiveDataQueued:  0000000000
    SendDataQueued:     0000000000
    SendStalled:        No
    SMC information:
      SMCStatus:        Inactive
      SMCReason:        00005303 - No active RNICs for the PNetID
    Ancillary Input Queue: N/A
    Application Data:   EZACICSO CSKL 0000038 CICSUSER CICP
```

```
----
Client Name: APPV4                    Client Id: 00000015
  Local Socket: 0.0.0.0..2049
  Foreign Socket: 9.42.103.99..1234
    BytesIn:            00000000000000000200
    BytesOut:           00000000000000000100
    DgramIn:            00000000000000000010
    DgramOut:           00000000000000000005
    StartDate:          06/17/2011        StartTime:          16:00:29
    Last Touched:       16:00:29
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicy:          Yes
      QOSRuleName:      QosRule2
    RoutingPolicy:      Yes
      RoutingTableName: prTab4
      RoutingRuleName:  SecLow4
    ReceiveDataQueued:  0000345655        ReceiveMsgCnt:      0000045644
      OldQDate:         09/15/06          OldQTime:           03:36:32
  Multicast Specific:
    TimeToLive:         0000000001        Loopback:  Yes
    OutgoingIpAddr:     199.1.2.3
    Group             IncomingIpAddr    SrcFltMd
    -----             --------------    --------
    224.8.8.8         193.1.1.94        Exclude
      SrcAddr: 20.20.20.20
               22.22.22.22
----
Client Name: APPV6                    Client Id: 00000016
  Local Socket: ::..2050
  Foreign Socket: 12AB::1..1235
    BytesIn:            00000000000000000200
    BytesOut:           00000000000000000100
    DgramIn:            00000000000000000010
    DgramOut:           00000000000000000005
    StartDate:          06/17/2011        StartTime:          16:00:29
    Last Touched:       16:00:29
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicy:          No
    RoutingPolicy:      No
    ReceiveDataQueued:  0000000000        ReceiveMsgCnt:      0000000000
  Multicast Specific:
    HopLimit:           0000000001        Loopback:  Yes
    OutgoingIntf:
    Group:              ff03::333
      IncomingIntf:     LINK6             SrcFltMd:  Exclude
        SrcAddr:        2e00::7
                        2e00::8
----
Client Name: SYSLOGD1                 Client Id: 0000002C
  Local Socket: 0.0.0.0..529
  Foreign Socket: *..*
    BytesIn:            00000000000000000000
    BytesOut:           00000000000000000000
    DgramIn:            00000000000000000000
    DgramOut:           00000000000000000000
    StartDate:          06/17/2011        StartTime:          14:27:42
    Last Touched:       14:27:42
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicy:          No
    RoutingPolicy:      No
    ReceiveDataQueued:  0000345655        ReceiveMsgCnt:      0000004564
      OldQDate:         09/15/06          OldQTime:           03:36:32
    ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
----
```

Report field descriptions:

- The following fields are displayed for a TCP connection entry:

Client Name

See the Client name or User ID information in "Netstat report general concept" on page 324 for a detailed description.

Client ID

See the Client ID or Connection Number information in "Netstat report general concept" on page 324 for a detailed description.

Local Socket

See the Local Socket information in "Netstat report general concept" on page 324 for a detailed description.

Foreign Socket

See the Foreign Socket information in "Netstat report general concept" on page 324 for a detailed description.

StartDate

Date of the last one of the following events that occurred for the TCP connection or UDP endpoint:

- UDP bind
- TCP bind
- TCP listen
- TCP connection establishment

StartTime

Time of the last one of the following events that occurred for the TCP connection or UDP endpoint:

- UDP bind
- TCP bind
- TCP listen
- TCP connection establishment

BytesIn

The number of bytes of data the stack has received for this connection. This includes both the total bytes that the application has received and the total bytes in the receive buffer that have not yet been read by the application.

Restriction: The TCP/IP stack maintains 64-bit counters for TCP connections and UDP endpoints. However, if you are running an IPv4-only stack, and the Netstat output is in the SHORT format, only the lower 32-bit counter value is displayed. If a large amount of data has been received, the number of bytes can exceed a 32-bit counter so the value displayed will appear to have been reset. Use the FORMAT/-M LONG output option on the Netstat command to cause Netstat to use the LONG format for the output. The LONG format displays the full 64-bit counter value. You can also specify the FORMAT parameter on the IPCONFIG profile statement to set FORMAT LONG as the default value for all Netstat commands.

BytesOut

The number of bytes of data the application has sent. This includes all the data that has been sent to the remote connection and all the data that has not been sent but is buffered and waiting to be sent by the local stack.

Restriction: The TCP/IP stack maintains 64-bit counters for TCP connections and UDP endpoints. However, if you are running an IPv4-only stack, and the Netstat output is in the SHORT format, only the lower 32-bit counter value is displayed. If a large amount of data has been sent, the number of bytes can exceed a 32-bit counter so the value displayed will appear to have been reset. Use the FORMAT/-M LONG output option on the Netstat command to cause Netstat to use the LONG format for the output. The LONG format displays the full 64-bit counter value. You can also specify the FORMAT parameter on the IPCONFIG profile statement to set FORMAT LONG as the default value for all Netstat commands.

SegmentsIn
The number of non-retransmitted TCP packets received for this connection.

Guideline: This value, when displayed for a TCP connection across an SMC-R link, includes the number of Remote Direct Memory Access (RDMA) inbound operations.

SegmentsOut
The number of non-retransmitted TCP packets sent for this connection.

Guideline: This value, when displayed for a TCP connection across an SMC-R link, includes the number of RDMA outbound operations.

Last touched
See the Last touched time information in “Netstat report general concept” on page 324 for a detailed description.

State Describes the state of the TCP connection. See “TCP connection status” on page 324 for more information.

RcvNxt
The sequence number of the next byte this side of the connection is expecting to receive. Each byte that is sent or received in a TCP connection has its own unique, ascending sequence number.

SndNxt
The sequence number of the next byte that the stack can send.

ClientRcvNxt
The sequence number of the next byte that the application will read from the receive buffer.

ClientSndNxt
The sequence number of the next byte of data that the application can add to the send buffer.

InitRcvSeqNum
The first sequence number that was received from the remote stack host when establishing the connection.

InitSndSeqNum
The first sequence number that the local stack sent out when establishing the connection.

CongestionWindow
The value that is used when congestion is detected in the network to limit the amount of data that is sent by the local stack. This value

represents the maximum amount of data that is sent without waiting for an acknowledgment from the remote socket.

SlowStartThreshold
The slow-start threshold is used to determine whether the connection is recovering from congestion. If the congestion window is smaller than the slow-start threshold, the connection will take actions to more quickly recover from congestion.

IncomingWindowNum
The incoming window number is the maximum sequence number that the remote socket can send until the local application reads more data from the local socket.

OutgoingWindowNum
The outgoing window number is the maximum sequence number that can be sent without waiting for the remote socket to read data (see the send window).

SndWl1
The sequence number from the segment that last updated the SndWnd field.

SndWl2
The acknowledgment number from the segment that last updated the SndWnd field.

SndWnd
The amount of available buffer space that is advertised by the remote side into which data can be sent.

MaxSndWnd
The largest send window the remote socket has sent to the local socket.

SndUna
This value is the sequence number of the first byte of data in the local socket's send buffer that has not been acknowledged by the remote socket.

rtt_seq
The sequence number of the byte of data sent in a packet for which the local socket is measuring the round-trip time (the time it takes between the local socket sending a packet and receiving an acknowledgment from the remote socket).

MaximumSegmentSize
The largest amount of data the local socket can send in a single packet.

DSField
The Differentiated Services Code Point value being used for this connection.

The DSField represents one of the following values:

- If there is a Service Policy Agent policy in effect for this entry, one of the following value is used:
  - The ToS value defined by RFC 791 and RFC 1349.
  - The Differentiated Services field value defined by RFC 2474.
- If there is no Service Policy Agent policy in effect for this entry, the value is 0.

Round-trip information

The round-trip time is the amount of time that elapses between the time a packet is sent and the time an acknowledgment for that packet is received.

Smooth trip time

The average amount of time it has taken for a packet to be sent and an acknowledgment to be received for this connection, measured in milliseconds.

SmoothTripVariance

The average variation in round-trip time, measured in milliseconds.

ReXmt

The total number of times a packet has been retransmitted for this connection. This count is historical for the life of the connection.

ReXmtCount

The number of times the last packet that was sent has been retransmitted.

DupACKs

The total number of duplicate acknowledgments that have been received by this connection.

RcvWnd

The amount of available buffer space that is advertised to the remote side into which data can be received.

SockOpt

Socket option flag. For TCP/IP stacks that are not IPv6 enabled, it is a one-byte hexadecimal value of common socket options. For IPv6-enabled TCP/IP stacks, it is a one-byte hexadecimal value of common socket options, followed by a one-byte hexadecimal value of IPv6-specific socket options.

Common socket options:

80 1... ....

Indicates that the socket option SO_REUSEADDR has been set for this socket. This socket option allows the socket to be bound to the same port that other sockets are bound to.

40 .1.. ....

Indicates that the socket option SO_OOBINLINE has been set for this socket. If this socket option is set, out-of-band data is returned in a normal read operation. If this socket option is not set, out-of-band data can be retrieved only by setting the MSG_OOB flag on a read operation.

20 ..1. ....

Indicates that the socket option SO_LINGER has been set for this socket. The SO_LINGER socket option allows an application to specify whether unsent data is discarded when the socket is closed, and how long to wait if the data is not discarded.

10 ...1 ....

Indicates that the socket option SO_DONTROUTE has been set for this socket. If this socket option is set, data is sent without regard to routes. This is equivalent to the MSG_DONTROUTE flag on a write operation.

08 .... 1...
Indicates the socket option TCP_NODELAY has been set for this socket. Unless this socket option is set, the TCP/IP stack will attempt to optimize the sending of small data packets by holding them briefly in case it has more data to send.

04 .... .1..
Indicates that the SO_KEEPALIVE socket option has been set for this socket. If this socket option is set, the TCP/IP stack will periodically send empty packets to the remote stack to make sure the connection is still alive.

IPv6 socket options:

80 1... ....
Indicates that the IPV6_UNICAST_HOPS option has been set for this socket.

20 ..1. ....
Indicates that the IPV6_USE_MIN_MTU for unicast option has been set for this socket.

10 ...1 ....
Indicates that the IPV6_TCLASS option has been set for this socket.

08 .... 1...
Indicates that the IPV6_RECVTCLASS option has been set for this socket.

04 .... .1..
Indicates that the IPV6_RECVHOPLIMIT option has been set for this socket.

02 .... ..1.
Indicates that the IPV6_V6ONLY option has been set for this socket.

Any other value
Used for diagnostic purposes only under the direction of IBM Service personnel.

TcpTimer
TCP timer flag. It is a one-byte hexadecimal value that is used for diagnostic purposes only under the direction of IBM Service personnel.

TcpSig
TCP signal flag. It is a one-byte hexadecimal value and can have one of the following values:

80 1... ...
Indicates the application has requested to receive the SIGURG signal when urgent data is received on this socket.

40 .1.. ....
Indicates the application has requested to receive the SIGIO signal when data is received on this socket.

Any other value
Is used for diagnostic purposes only under the direction of IBM Service personnel.

TcpSel TCP select flag. It is a one-byte hexadecimal value that is used for diagnostic purposes only under the direction of IBM Service personnel.

TcpDet

Special TCP protocol flag. It is a one-byte hexadecimal value:

04 .... .1..

Indicates the TCP_KEEPALIVE socket option has been set for this socket. This socket option is used to set a socket-specific time interval value for use with the SO_KEEPALIVE socket option. See the description of field SockOpt for an explanation of the SO_KEEPALIVE socket option. The TCP_KEEPALIVE time interval value is in effect only if the SO_KEEPALIVE socket option is set for the socket.

Any other value

Is used for diagnostic purposes only under the direction of IBM Service personnel.

TcpPol

TCP poll flag. It is a one-byte hexadecimal value to be used for diagnostic purposes only under the direction of IBM Service personnel.

TcpPrf A 1-byte hexadecimal TCP performance flag that can have the following values:

40 .1.. ....

Indicates that Dynamic Right Sizing (DRS) is active for this connection so the stack is automatically tuning the advertised receive window. For more information about DRS, see TCP receive window in z/OS Communications Server: IP Configuration Guide. The RcvWnd field shows the current size of the receive window for this connection.

02 .... ..1.

Indicates that DRS was active for this connection, but has been disabled. This is caused by the associated application not reading the data as fast as the data arrives.

Any other value

Used for diagnostic purposes only under the direction of IBM Service personnel.

QOSPolicy

Indicates whether a matching QoS policy rule has been found for this connection. This field can have the following values:

`No` Indicates that a matching QoS policy rule was not found for this connection.

`Yes`

Indicates that a matching QoS policy rule was found for this connection. When the QOSPolicy field has the value `Yes`, the following information is displayed:

`QOSRuleName`

The name of the Policy rule that is in use for this connection. This policy is for outbound traffic only.

TTLSPolicy

Indicates whether a matching Application Transparent Transport Layer Security (AT-TLS) policy rule has been found for this connection. This set of fields is not displayed if the AT-TLS function was disabled when the connection was established (NOTTLS was specified on the TCPCONFIG statement or is in effect by default) or policy lookup has not yet occurred.

– **TTLSPolicy: No** indicates that no matching AT-TLS policy rule was found for this connection. There is no rule or action listed.

– **TTLSPolicy: Yes** indicates one of the following case:

  - A matching AT-TLS policy rule was found for this connection with an indication that AT-TLS should be enabled (TTLSEnabled ON was specified on the TTLSGroupAction). The rule and actions are displayed.
  - A matching AT-TLS policy rule was found for this connection with an indication that AT-TLS should be disabled (TTLSEnabled OFF was specified on the TTLSGroupAction). The rule and actions are displayed.

TTLSRule
The name of the AT-TLS policy rule that is in use for this connection, followed by `(Stale)` when the rule is no longer available for use by new connections. This field is not displayed when the connection does not match a policy rule.

TTLSGrpAction
The name of the AT-TLS policy group action that is in use for this connection, followed by `(Stale)` when the action is no longer available for use by new connections. This field is not displayed when the connection does not match a policy rule.

TTLSEnvAction
The name of the AT-TLS policy environment action that is in use for this connection, followed by `(Stale)` when the action is no longer available for use by new connections. This field is not displayed when the connection does not match a policy rule or when no TTLSEnvironmentAction was specified.

TTLSConnAction
The name of the AT-TLS policy connection action that is in use for this connection, followed by `(Stale)` when the action is no longer available for use by new connections. This field is not displayed when the connection does not match a policy rule or when no TTLSConnectionAction was specified.

RoutingPolicy
Indicates whether a matching routing policy rule has been found for this connection. This field can have the following values:

No Indicates that no matching routing policy rule was found for this connection.

For an Enterprise Extender (EE) UDP socket entry, the RoutingPolicy value is always `No`. Display the routing policy information for an Enterprise Extender (EE) UDP socket entry by using the DISPLAY NET,EEDIAG,TEST=YES command. See z/OS Communications Server: SNA Operation for details.

Yes Indicates that a matching routing policy rule was found for this connection.

When the RoutingPolicy value is Yes, the following information is displayed:

RoutingTableName
The name of the routing table that was used to find the route for this connection or *NONE* if a route was not found. The value EZBMAIN is displayed when the main routing table was used.

RoutingRuleName
The name of the routing policy rule in use for this connection.

ReceiveBufferSize
The number of bytes received from the remote application that this connection is allowed to maintain in a buffer. All the data that is received is kept in a buffer until the local application reads the data.

SendBufferSize
The number of bytes the local application has sent that this connection is allowed to maintain in a buffer. All data that the application has sent is kept in the buffer until the remote side acknowledges receiving the sent data.

TcpClusterConnFlag
TCP cluster connection type flag. It is a one-byte hexadecimal field and can have one of the following values:

80 1... ....
Indicates that the SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO ioctl was requested.

08 .... 1...
If the SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO ioctl was issued for this socket, this bit indicates that the communication from this node to the stack hosting the partner application is not sent on links/interfaces exposed outside the cluster (sysplex).

04 .... .1..
If the SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO ioctl was issued for this socket, this bit indicates that the connection partners are in the same MVS image.

02 .... ..1.
If the SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO ioctl was issued for this socket, this bit indicates that the connection partners are in the same cluster.

01 1
If the SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO ioctl was issued for this socket, this bit indicates that the connection partners are not in the same cluster.

00 ....
If the TcpTrustedPartner flag indicates that the SIOCSPARTNERINFO ioctl has been successfully issued or inherited from the listener socket, this value indicates that the SO_CLUSTERCONNTYPE socket option or the SIOCGPARTNERINFO ioctl has not been issued for this socket.

Any other value
Used for diagnostic purposes only under the direction of IBM Service personnel.

For more information about the cluster connection type, see the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference. For more information about the SIOCGPARTNERINFO ioctl, see z/OS Communications Server: IP Programmer's Guide and Reference.

TcpTrustedPartner
The TCP trusted connection flag is displayed in the following situations:

- Security credentials of a partner within a sysplex or subplex have been retrieved over a trusted TCP connection using the SIOCGPARTNERINFO ioctl.
- The SIOCSPARTNERINFO ioctl has been issued for the socket.

The TCP trusted connection flag is a 1-byte hexadecimal field and can have the following values:

80 1... ....
This bit indicates that the partner address-space user ID has been retrieved, as well as the task-level user ID if it is available.

40 .1.. ....
This bit indicates that the partner address-space UTOKEN has been retrieved, as well as the task-level UTOKEN if it is available.

20 ..1. ....
This bit indicates that the SIOCSPARTNERINFO ioctl has been successfully issued or inherited from the listener socket.

For information about trusted TCP/IP connections and the SIOCGPARTNERINFO and SIOCSPARTNERINFO ioctl calls, see z/OS Communications Server: IP Programmer's Guide and Reference.

ReceiveDataQueued
The number of bytes of data on the receive queue from the remote application yet to be read. This field is not displayed for a connection that is in listen state. The amount of data queued can be up to double the ReceiveBufferSize size. When the number of bytes is not zero, the following information is displayed:

OldQDate
The date of the oldest data on the receive queue.

OldQTime
The time of the oldest data on the receive queue. This value does not include leap seconds.

The ReceiveDataQueued information is not displayed for a connection that is in listen state.

SendDataQueued
The number of bytes of data on the send queue waiting for the remote side to acknowledge. This field is not displayed for a connection that is in listen state. The amount of data queued can be up to double the size of the SendBufferSize. When the number of bytes is not zero, the following information is displayed:

OldQDate
The date of the oldest data on the send queue.

OldQTime
The time of the oldest data on the send queue. This value does not include leap seconds.

The SendDataQueued information is not displayed for a connection that is in listen state.

SendStalled
Indicates whether this connection's send data flow is stalled. The send data flow is considered stalled if one or more of the following conditions are true:

- The TCP send window size is less than 256 or is less than the smaller of the largest send window that has been seen for the connection and the default MTU. The TCP send window size is set based on values provided by the TCP peer. The default MTU for IPv4 is 576. The default MTU for IPv6 is 1280.
- The TCP send queue is full and the data is not being retransmitted.

This field is not displayed for a connection that is in listen state. If the value is Yes, then this connection's send data flow is stalled.

SMC Information
The SMC information. This section is displayed for connections when at least one Peripheral Component Interconnect Express (PCIe) function ID (PFID) was defined by using the SMCR parameter of the GLOBALCONFIG statement. The `SMC Information` section contains the following information:

SMCStatus
Indicates whether this connection is traversing a Shared Memory Communications over Remote Direct Memory Access (SMC-R) link. This field can have the following values:

Inactive
Indicates that this connection does not use an SMC-R link.

When the `SMCStatus` value is `Inactive`, the following information is displayed:

SMCReason *reasonCode - reasonText*
This field provides an explanation for why the connection is not using an SMC-R link. The reason code and text can be one of the following values:

5013 - RDMA connectivity failure
SMC-R communications cannot be used for this connection because the first attempt to send data over RDMA encountered an error. A likely reason for this error is a configuration problem in the switch that is connected to the RoCE Express interface. For example, an incorrect VLANID value was configured on the switch port for the RoCE Express interface.

5203 - Insufficient virtual storage
SMC-R communications cannot be used for this connection because TCP private 64-bit virtual storage could not be allocated for an RMB buffer.

5204 - SMCR FIXEDMemory limit exceeded reached
SMC-R communications cannot be used for this connection because the required SMC-R memory could not be allocated.

5205 - TCP connection limit reached
SMC-R communications cannot be used for this connection because another RMB for a new connection could not be obtained.

5206 - VLAN ID not found
SMC-R communications cannot be used for this connection because no VLAN that was enabled by SMC-R was found.

5209 - No qualifying active RNICs
No active IBM 10GbE RoCE Express interfaces are detected in the SMC-R layer that can be used for this TCP connection.

5219 - Peer is out of synch
SMC-R communications cannot be used for this connection because the peer is out of synchronization condition during negotiation.

521E - Peer subnet/prefix mismatch
SMC-R communications cannot be used for this connection because the peer does not have an active interface in the same subnet that is eligible for SMC-R.

5301 - Peer did not accept SMC-R request
The remote connection peer is not configured to use SMC-R communications.

5302 - Route not SMC-R eligible
SMC-R communications cannot be used for this connection because of connectivity issues or the absence of an active interface that supports SMC-R processing.

5303 - No active RNICs for the PNetID
No active 10GbE RoCE Express features are detected for the PNetID.

5304 - Connection is local
The connection peers are on the same TCP/IP stack.

5306 - No storage for SMC-R negotiation
Storage for SMC-R negotiation over this TCP connection cannot be obtained.

5307 - Connection uses lPSec
SMC-R communications cannot be used for this connection because the connection is using IP security.

5308 - FRCA server
SMC-R communications cannot be used for this connection because the connection is used by a Fast Response Cache Accelerator (FRCA) server.

5309 - Pascal application
SMC-R communications cannot be used because the connection is used by a Pascal API application.

530A - NOSMCR Port server
SMC-R communications cannot be used for this connection because the server port was configured with the NOSMCR option.

530B - Invalid MTU from peer
SMC-R communications cannot be used for this connection because the peer had an invalid MTU size for this SMC-R link.

530C - No prefix on interface
SMC-R communications could not be used for this connection because of no valid IPv6 prefixes for the associated OSD interface.

reasonCode **- Internal error**
SMC-R communications cannot be used for this connection because of an internal error.

reasonCode **- *Peer generated***
SMC-R communications cannot be used for this connection because the peer reported an error. See the peer product's documentation for additional details.

Active Indicates that this connection uses an SMC-R link.

When the `SMCStatus` value is `Active`, the following information is displayed:

SMCGroupId
This field identifies the SMC-R link group that includes the individual SMC-R link that this connection traverses. This TCP/IP stack generates the SMC-R link group identifier dynamically.

LocalSMCLinkId
This field identifies the SMC-R link on this

TCP/IP stack that this connection traverses. This TCP/IP stack generates the SMC-R link identifier dynamically.

RemoteSMCLinkId
This field identifies the SMC-R link on the remote peer that this connection traverses. The remote peer generates this SMC-R link identifier and provides it to this TCP/IP stack during SMC-R link activation.

LocalSMCRcvBuf
This field indicates the size of the RMB element that the local host uses for receiving data on this connection from the remote host.

RemoteSMCRcvBuf
This field indicates the size of the RMB element that the remote host uses for receiving data on this connection from the local host.

Ancillary Input Queue
Indicates whether this connection is registered to the TCP bulk data ancillary input queue. This field is not displayed for a connection that is in listen state. This field can have the following values:

N/A Indicates that this connection is not registered to the TCP bulk data ancillary input queue.

Yes Indicates that this connection is registered to the TCP bulk data ancillary input queue.

When the Ancillary Input Queue value is Yes, the following information is displayed:

BulkDataIntfName
This field indicates the name of the interface over which the inbound traffic is being received.

ConnectionsIn
The number of connections that a server has accepted. This field is displayed only for a connection that is in listen state. Once a connection has been accepted, communication can begin between the client and server applications.

ConnectionsDropped
The number of connection requests that have been received by the server and dropped because the maximum number of connection requests was already in the backlog queue. This field is displayed only for a connection that is in listen state.

MaximumBacklog
The maximum number of connections that a server maintains on the backlog queue. This field is displayed only for a connection that is in listen state. Connection requests that are received when the maximum number of connections requests is already on the backlog queue are typically discarded. A high maximum backlog queue value causes more simultaneous connection requests than a server can handle without having to drop requests.

ConnectionFlood
Indicates whether this server is experiencing a potential connection flood

attack. A server is considered under a potential connection flood attack when backlog queue expansion is required to handle the incoming connection requests. The point where a potential connection flood attack is detected is based on the initial size of the backlog queue. A small initial backlog queue (for example, 10 entries) is allowed to expand twice before the server is considered under attack, while a server with a large initial backlog queue (for example, 500 entries) can expand once, up to a maximum of 768 entries, before it is considered under attack. This field is displayed only for a connection that is in listen state. If the value is Yes, then this server is experiencing a potential connection flood attack.

CurrentBacklog

The number of connections that are currently in the backlog queue. This field is displayed only for a connection that is in listen state. This value includes connections that are fully established and that are ready to be accepted by the server application; it also includes connections that are not yet fully established (the TCP connection establishment handshake is not yet complete). To determine the number of connections in the backlog queue that are not fully established, subtract the ServerBacklog value from the CurrentBacklog value. If the server application uses the Fast Response Cache Accelerator (FRCA) feature, fully established connections that are being serviced by TCP/IP from the FRCA cache are also included in the CurrentBacklog value. The FRCABacklog value in this report indicates the number of these connections.

ServerBacklog

The number of connections currently in the backlog queue that are established and that have not yet been accepted.

FRCABacklog

The number of connections currently in the backlog queue that are established FRCA connections and that are being serviced by TCP/IP from the FRCA cache. These connections do not need to be accepted by the server application. This field is applicable only for server applications that use the FRCA feature.

CurrentConnections

The number of currently established connections to the server. This field is displayed only for a connection that is in listen state.

SEF

The server accept efficiency fraction (SEF) is a measure, calculated at intervals of approximately one minute, of the efficiency of the server application in accepting new connection setup requests and managing its backlog queue. The value is displayed as a percentage. A value of 100 indicates that the server application is successfully accepting all its new connection setup requests. A value of 0 indicates that the server application is not responding to new connection setup requests. This field is displayed only for a connection that is in listen state.

When using SHAREPORTWLM, the SEF value is used to modify the WLM server-specific weights, thereby influencing how new connection setup requests are distributed to the servers sharing this port. When using SHAREPORT, the SEF value is used to weight the distribution of new connection setup requests among the SHAREPORT servers. Whether SHAREPORT or SHAREPORTWLM are specified, the SEF value is reported back to the distributor to be used as part of the target server responsiveness fraction calculation, which influences how new connection setup requests are distributed to the target servers.

Quiesced

Indicates whether this server application has been quiesced for DVIPA sysplex distributor workload balancing. This field is displayed only for a connection that is in listen state. If the value is Dest, then this server will receive no new DVIPA sysplex distributor workload connections until the server application has been resumed. When the server application is resumed, the quiesced value changes to No.

SharePort

Indicates that multiple TCP listening servers are sharing the same port. This field is displayed only for a connection that is in listen state. The method used by TCP to distribute incoming connections to the listeners is indicated by Base or WLM described below. See the PORT profile statement in the z/OS Communications Server: IP Configuration Reference for more information on sharing a TCP port.

Base Connections are proportionally distributed among the available shareport listeners using the SEF value. This value corresponds to the SHAREPORT parameter on the PORT profile statement.

WLM Connections are distributed among the available shareport listeners using the normalized WLM server-specific weights. This value corresponds to the SHAREPORTWLM parameter on the PORT profile statement.

RawWeight

The raw composite weight for this server. The composite weight is based on the application's general CPU, zAAP, and zIIP processor utilization.

NormalizedWeight

The normalized values of the WLM server-specific weights. The original raw weights received from WLM are proportionally reduced for use by the distribution algorithm. Connections are distributed to these servers in a weighted round-robin fashion using the normalized weights if SHAREPORTWLM is specified on the PORT profile statement. The displayed normalized weight is shown after it has been modified by the SEF value. This field is shown regardless of the distribution method (Base or WLM) that is used.

Abnorm

Indicates whether the server application is experiencing conditions that cause transactions to complete abnormally. The value represents a rate of abnormal transaction completions per 1000 total transaction completions. It is applicable only for TCP applications that act as Subsystem Work Managers and report transaction status using Workload Management Services, such as IWMRPT. For example, the value 100 indicates that 10% of all transactions processed by the server application are completing abnormally. Under normal conditions, this value is 0. A nonzero value indicates that the server application has reported some abnormal transactions completions to WLM and that WLM has reduced the recommendation provided to sysplex distributor for this server instance. This reduction in the WLM recommendation enables more new TCP connections to be directed to servers that are not experiencing problem conditions that lead to abnormal transaction completions.

The greater the Abnorm rate field value, the greater the reduction WLM applies to the recommendation for this target instance. For more information about the conditions that cause the abnormal transaction completions for a given server application, see the documentation provided by the server application.

If applications do not provide this transaction status to WLM or SHAREPORTWLM is not configured, then this field has the value 0. For more information about workload management interfaces, see z/OS MVS Programming: Workload Management Services.

Health

The server application health indicator. This health indicator is available only for applications that provide this information to WLM using the IWM4HLTH or IWMSRSRG services. It provides a general health indication for an application or subsystem. Under normal circumstances, the value of this field is 100, indicating that the server is 100% healthy. Any value that is less than 100 indicates that the server is experiencing problem conditions that might prevent new work requests from being successfully processed. A value of less than 100 also causes the WLM to reduce the recommendation provided to the sysplex distributor for this server instance. This reduction in the WLM recommendation enables more new TCP connections to be directed to servers that are not experiencing problem conditions.

The reduction in the WLM recommendation is proportional to value of the Health indicator. For example, if the health value is 20%, WLM reduces the recommendation for this server by 80%. For more information about the conditions leading to a health indicator of less than 100, see the documentation for the server application.

If applications do not provide this health indicator to WLM or SHAREPORTWLM is not configured, then the value of this field is 100. For more information about workload management interfaces, see z/OS MVS Programming: Workload Management Services.

RawCP

The raw WLM server-specific general CP weight.

RawzAAP

The raw WLM server-specific zAAP weight.

RawzIIP

The raw WLM server-specific zIIP weight.

PropCP

The RawCP value modified by the proportion of CP capacity that is currently being consumed by the application's workload as compared to the other processors (zIIP and zAAP).

PropzAAP

The RawzAAP value modified by the proportion of zAAP capacity that is currently being consumed by the application's workload as compared to the other processors (CP and zIIP).

PropzIIP
The RawzIIP value modified by the proportion of zIIP capacity that is currently being consumed by the application's workload as compared to the other processors (CP and zAAP).

ILWeighting
The weighting factor the workload manager (WLM) uses when it compares displaceable capacity at different importance levels (ILs) in order to determine a SERVERWLM recommendation for each system.

XcostzAAP
The crossover cost that is applied to the workload that was targeted to run on a zAAP processor but that ran on the conventional processor.

XcostzIIP
The crossover cost that is applied to the workload that was targeted to run on a zIIP processor but that ran on the conventional processor.

Application Data
The application data that makes it easy for users to locate and display the connections that are used by the application. The beginning of the application data identifies the format of the application data area. For z/OS Communications Server applications, see application data in the z/OS Communications Server: IP Programmer's Guide and Reference for a description of the format, content, and meaning of the data supplied by the application. For other applications, see the documentation that is supplied by the application. The data is displayed in character format if application data is present. Non-printable characters, if any, are displayed as dots.

- The following fields are displayed for a UDP socket entry:

Client Name
See the Client name or User ID information in “Netstat report general concept” on page 324 for a detailed description.

Client ID
See the Client ID or Connection Number information in “Netstat report general concept” on page 324 for a detailed description.

Local Socket
See the Local Socket information in “Netstat report general concept” on page 324 for a detailed description.

Foreign Socket
See the Foreign Socket information in “Netstat report general concept” on page 324 for a detailed description.

BytesIn
The number of bytes of data the stack has received for this UDP socket. Includes both the total bytes that all applications have received for this socket and the total bytes in stack buffers that have not yet been read by any application.

BytesOut
Number of outbound bytes of user data sent from this socket.

DgramIn
The number of datagrams the stack has received for this UDP socket.

This includes both the total datagrams that all applications have received for this socket and the total datagrams in stack buffers that have not yet been read by any application. A datagram is the group of data bytes contained in a UDP packet.

DgramOut
Number of outbound datagrams sent from this socket.

Last touched time
See the Last touched time information in "Netstat report general concept" on page 324 for a detailed description.

MaxSendLim
Maximum allowed size of a user datagram sent from this socket.

MaxRecvLim
Maximum allowed size of a user datagram received on this socket.

SockOpt
Socket option flag. For TCP/IP stacks that are not IPv6 enabled, it is a one-byte hexadecimal value of common socket options. For IPv6-enabled TCP/IP stacks, it is a one-byte hexadecimal value of common socket options, followed by a three-byte hexadecimal value of IPv6-specific socket options.

IPv4 socket options:

80 1... ....
Allow use of broadcast address (IPv4 only)

40 .1.. ....
Allow loopback of datagrams

20 ..1. ....
Bypass normal routing

10 ...1 ....
Forward ICMP messages (Pascal API)

08 .... 1...
Last sent a multicast packet

04 .... .1..
Multicast packets can be received by this socket

02 .... ..1.
Reuse address

other values
reserved

IPv6 socket options:

Byte 1

80 1... ....
AF_INET6 socket

40 .1.. ....
IPV6_V6ONLY option set

20 ..1. ....
IPV6_RECVPKTINFO option set

10 ...1 ....
IPV6_RECVHOPLIMIT option set

08 .... 1...
IPV6_USE_MIN_MTU for unicast option

04 .... .1..
IPV6_PKTINFO src IP@ option set

02 .... ..1.
IPV6_PKTINFO interface index option set

01 1
IPV6_UNICAST_HOPS option set

Byte 2

80 1... ....
IPV6_USE_MIN_MTU for multicast option set

40 .1.. ....
IPV6_RECVRTHDR option set

20 ..1. ....
IPV6_RECVHOPOPTS option set

10 ...1 ....
IPV6_RECVDSTOPTS option set

08 .... 1...
IPV6_RECVTCLASS option set

04 .... .1..
IPV6_NEXTHOP option set

02 .... ..1.
IPV6_RTHDR option set

01 1
IPV6_HOPOPTS option set

Byte 3

80 1... ....
IPV6_DSTOPTS option set

40 .1.. ....
IPV6_RTHDRDSTOPTS option set

20 ..1. ....
IPV6_TCLASS option set

10 ...1 ....
IPV6_DONTFRAG option set

08 .... 1...
IPV6_RECVPATHMTU option set

other values
reserved

DSField
The Differentiated Services Code Point value being used for this connection.

The DSField represents one of the following values:

– If there is a Service Policy Agent policy in effect for this entry, one of the following value is used:
  - The ToS value defined by RFC 791 and 1349

- The Differentiated Services field value defined by RFC 2474

– For UDP entries for which there is no Service Policy Agent policy in effect but the entry is being used for an Enterprise Extender connection, the hexadecimal value of one of the following VTAM IP Type of Service values is displayed:

20	Low
40	Medium
80	High
C0	Network

See the z/OS Communications Server: SNA Network Implementation Guide for additional information.

– If neither of these is true, this value is 0.

QOSPolicy

Indicates whether a matching QoS policy rule has been found for this connection. This field can have the following values:

No Indicates that a matching QoS policy rule was not found for this connection.

Yes

Indicates that a matching QoS policy rule was found for this connection. When the QOSPolicy field has the value Yes, the following information is displayed:

QOSRuleName

The name of the Policy rule that is in use for this connection. This policy is for outbound traffic only.

RoutingPolicy

Indicates whether a matching routing policy rule has been found for this connection. This field can have the following values:

No Indicates that no matching routing policy rule was found for this connection.

Yes Indicates that a matching routing policy rule was found for this connection.

When the RoutingPolicy field has the value Yes, the following information is displayed:

RoutingTableName

The name of the routing table that was used to find the route for this connection or *NONE* if a route was not found. The value EZBMAIN is displayed when the main routing table was used.

RoutingRuleName

The name of the routing policy rule in use for this connection.

ReceiveDataQueued

The number of bytes of data on the receive queue from the remote application yet to be read. When the number of bytes is not zero, the following information is displayed:

OldQDate

The date of the oldest datagram on the receive queue.

OldQTime
The time of the oldest datagram on the receive queue.

ReceiveMsgCnt
The number of datagrams on the receive queue.

Multicast Specific
Indicates that there is multicast data associated with this socket.

For outgoing multicast data the following field descriptions apply:

HopLimit
The time-to-live value.

LoopBack
Indicates whether datagrams are sent to loopback.

OutgoingIpAddr
The IPv4 IP address of the link on which the datagrams are sent. The value of this field is 0.0.0.0 if the socket has not been set with the IP_MULTICAST_IF setsockopt option. This field is not applicable for an IPv6 multicast entry.

OutgoingIntf
The IPv6 interface name on which the datagrams are sent. The value of this field is blank if the socket has not been set with the IPV6_MULTICAST_IF setsockopt option. This field is not applicable for an IPv4 multicast entry.

For incoming multicast data the following field descriptions apply:

Group The multicast IP addresses (up to a maximum of 20) for which data is being received.

IncomingIpAddr
The IPv4 IP address of the link over which multicast datagrams are accepted. This field is not applicable for an IPv6 multicast entry.

IncomingIntf
The IPv6 interface name over which multicast datagrams are accepted. This field is not applicable for an IPv4 multicast entry.

SrcFltMd
The source filter mode, which can have a value of either `Include` or `Exclude`. A source filter applies only to incoming multicast data. This source filter function is set by an application for the UDP socket. See the information about Designing multicast programs in the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference for details. The source filter applies to all the IP addresses in the SrcAddr fields for the associated IncomingIPAddr address or IncomingIntf interface.

Include
Indicates that the socket receives only multicast datagrams that have a source IP address that matches an IP address indicated in the SrcAddr field.

Exclude

Indicates either that the source filter function is not active for the socket or that the application has requested to receive only multicast datagrams that have a source IP address that does not match an IP address indicated in the SrcAddr field. If the source filter function is not active or if the source filter function is active but no SrcAddr value is set, then the SrcAddr field contains the value `None`.

SrcAddr

Source address information for the socket.

ipaddr The source IP addresses (up to a maximum of 64), used in conjunction with the SrcFltMd value, that is used to determine which incoming multicast datagrams should be passed to an application.

None This value is displayed only when the source filter function is not active for the socket or when no source IP address is associated with group multicast address, IncomingIPAddr address, or IncomingIntf interface. The value of the corresponding SrcFltMd field is `Exclude`.

StartDate

See the StartDate information in "Netstat report general concept" on page 324.

StartTime

See the StartTime information in "Netstat report general concept" on page 324.

Netstat ALLConn/-a report

Provides information for all TCP connections and UDP sockets, including recently closed ones.

TSO syntax:

```
►►──NETSTAT ALLConn──┤ Modifer ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────►◄
```

Modifier:

```
►►──APPLDATA─────────────────────────────────────────────────────────────────►◄
```

APPLDATA

Provides application data in the output report.

Target:

Provide the report for a specified TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:

```
►►──┬─APPLD──appldata───────────────────────────────┬──────────────────►◄
    │            ┌──────────────┐                    │
    │            ▼              │                    │
    ├─CLIent────────clientname──┴────────────────────┤
    ├─HOSTName──hostname─────────────────────────────┤
    │            ┌──────────────────────────┐        │
    │            ▼                          │        │
    ├─IPAddr──────┬─ipaddr───────────────┬──┴────────┤
    │             ├─ipaddr/prefixLen─────┤           │
    │             └─ipaddr/subnetmask────┘           │
    │            ┌──────────────────┐                │
    │            ▼                  │                │
    ├─IPPort──────ipaddr+portnum────┴────────────────┤
    ├─NOTN3270───────────────────────────────────────┤
    │          ┌──────────────┐                      │
    │          ▼              │                      │
    ├─POrt────────portnum─────┴──────────────────────┤
    ├─SMCID──┬─smcid─┬───────────────────────────────┤
    │        └─*─────┘                               │
    └─CONNType──┬─NOTTLSPolicy────────────────────┬──┘
                └─TTLSPolicy──┬─────────────────┬─┘
                              ├─CURRent─────────┤
                              ├─GRoup──groupid──┤
                              └─STALE───────────┘
```

z/OS UNIX syntax:

```
►►──netstat  -a──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├─────────────►◄
```

Modifier:

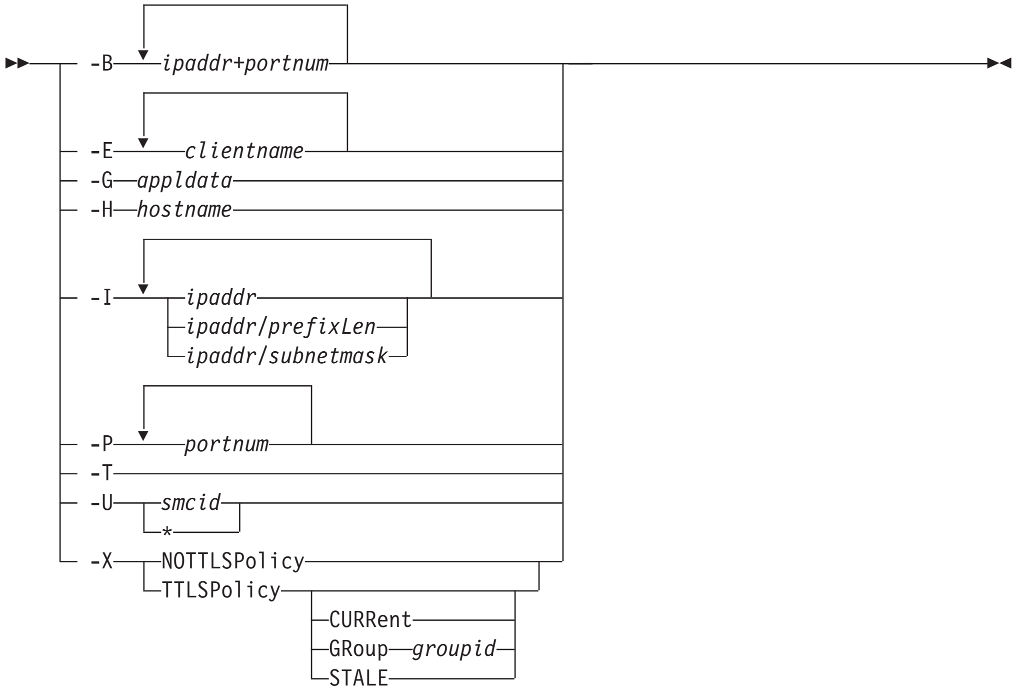
```
►►──APPLDATA─────────────────────────────────────────────────────────────────►◄
```

APPLDATA
Provides application data in the output report.

Target:
Provide the report for a specified TCP/IP address space by using **-p** *tcpname*. See “The Netstat command target” on page 316 for more information about the **-p** parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Filter:



Filter description:

APPLD/-G *appldata*

Filter the output of the ALLConn/**-a** report using the specified application data *appldata*. You can enter one filter value at a time and the specified value can be up to 40 characters in length.

CLIent/-E *clientname*

Filter the output of the ALLConn/**-a** report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be up to eight characters in length.

HOSTName/-H *hostname*

Filter the output of the ALLConn/**-a** report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters in length.

Result: At the end of the report, Netstat displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the ALLConn/**-a** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength

Filter the output of the ALLConn/**-a** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the ALLConn/**-a** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be either the local or remote IP address.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and will usually provide the same result as its IPv4 address.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

IPPort/-B *ipaddr***+***portnum*

Filter the report output of the ALLConn/**-a** report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be either the local or remote IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.

- For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.

NOTN3270/-T
Filter the output of the ALLConn/**-a** report, excluding TN3270 server connections.

POrt/-P *portnum*
Filter the output of the ALLConn/**-a** report using the specified port number *portnum*. You can enter up to six filter values.

Guideline: The port number can be either a local or remote port.

Restriction: For a UDP endpoint socket, the filter value applies only to the local or source port.

SMCID/-U *smcid*
Filter the output of the ALLConn/-a report by using the specified Shared Memory Communications over Remote Direct Memory Access (SMC-R) link or link group identifier *smcid*. If an asterisk (*) is specified for the filter value, Netstat provides output only for entries that are associated with SMC-R link, and link groups. You can enter one filter value at a time.

CONNType/-X
Filter the report using the specified connection type. You can enter one filter value at a time.

NOTTLSPolicy
Filter the output of the ALLConn/**-a** report, displaying only connections that have not been matched to an Application Transparent Transport Layer Security (AT-TLS) rule. This includes connections that were established while the AT-TLS function was disabled (the value NOTTLS was specified on the TCPCONFIG statement or is in effect by default) and all connections that are not TCP protocol. For TCP connections that were established while the AT-TLS function was enabled, this includes the following information:

- Connections for which AT-TLS policy lookup has not yet occurred (typically the first send or receive has not been issued yet)
- Connections for which AT-TLS policy lookup has occurred but no matching rule was found

TTLSPolicy
Filter the output of the ALLConn/**-a** report, displaying only connections that match an Application Transparent Transport Layer Security (AT-TLS) rule. This includes only connections that were established while the AT-TLS function was enabled, for which an AT-TLS policy rule was found that has the value `TTLSEnabled ON` or `TTLSEnabled OFF` specified in the TTLSGroupAction policy statement. Responses can be further limited on AT-TLS connection type. AT-TLS connection type has the following values:

CURRent
Display only connections that are using AT-TLS where the rule and all actions are still available to be used for new connections.

GRoup *groupid*
Display only connections that are using the AT-TLS group

specified by the *groupid* value. The specified *groupid* value is a number that is assigned by the TCP/IP stack to uniquely identify an AT-TLS group. You can determine the *groupid* value from the GroupID field in the Netstat TTLS/**-x** GROUP report.

STALE

Display only connections that are using AT-TLS where the rule or at least one action is no longer available to be used for new connections.

The filter value for CLIent/**-E**, IPAddr/**-I**, and APPLD/-G can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string *searchee* matches with **ar?he**, but the string *searhee* does not match with **ar?he**. If you want to use the wildcard character on the IPAddr/**-I** filter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/prefixlen* or *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -a -I '10.*.0.0'**.

Command syntax examples:

From TSO environment:

```
NETSTAT ALLCONN
   Display information for all TCP connections and UDP sockets, including recently closed
   ones in the default TCP/IP stack.
NETSTAT ALLCONN TCP TCPCS6
   Display information for all TCP connections and UDP sockets, including recently closed
   ones in TCPCS6 stack.
NETSTAT ALLCONN TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display information for these TCP connections and UDP sockets, including recently closed
   ones in TCPCS8 stack whose local or remote IP addresses match the specified filter IP
   address values.
NETSTAT ALLCONN (PORT 2222 6666 88
   Display information for those TCP connections and UDP sockets, including recently closed
   ones in the default TCP/IP stack whose local or remote ports match the specified filter
   port numbers.
```

From UNIX shell environment:

```
   netstat -a
   netstat -a -p tcpcs6
   netstat -a -p tcpcs6 -I 9.43.1.1 9.43.2.2
   netstat -a -P 2222 6666 88
```

Report examples:
The following examples are generated using the TSO NETSTAT command. The z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT ALLCONN
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
User Id  Conn     Local Socket           Foreign Socket         State
-------  ----     ------------           --------------         -----
FTPD1    0000003B 0.0.0.0..21            0.0.0.0..0             Listen
FTPD1    0000003D 9.37.65.146..21        9.67.115.5..1026       Establsh
FTPD1    0000003F 9.37.65.146..21        9.27.13.21..3711       Establsh
TCPCS    0000000F 0.0.0.0..23            0.0.0.0..0             Listen
TCPCS    0000000C 9.67.115.5..23         9.27.11.182..4886      Establsh
USER1    00000027 9.67.115.67..1027      9.67.115.5..21         ClosWait
USER1    00000029 9.67.115.69..1028      9.67.115.5..20         ClosWait
APPV4    00000015 0.0.0.0..2049          9.42.103.99..1234      UDP
SYSLOGD1 00000010 0.0.0.0..514           *..*                   UDP
```

IPv6 enabled or request for LONG format:

```
NETSTAT ALLCONN
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
User Id  Conn     State
-------  ----     -----
FTPD1    0000004A Listen
  Local Socket:   ::..21
  Foreign Socket: ::..0
FTPD1    00000052 Establsh
  Local Socket:   ::ffff:9.67.115.5..21
  Foreign Socket: ::ffff:9.67.115.65..1026
FTPD1    00000058 Establsh
  Local Socket:   2001:0db8::9:67:115:66..21
  Foreign Socket: 2001:0db8::9:67:115:65..1027
TCPCS    0000001A Listen
  Local Socket:   0.0.0.0..23
  Foreign Socket: 0.0.0.0..0
TCPCS    0000001E Establsh
  Local Socket:   9.67.115.5..23
  Foreign Socket: 9.27.11.182..4665
USER3    0000005F Establsh
  Local Socket:   2001:0db8::9:67:115:5..1079
  Foreign Socket: 2001:0db8::9:67:115:65..21
USER6    000000C7 Establsh
  Local Socket:   9.67.115.5..1027
  Foreign Socket: 9.37.65.146..21
USER8    000000B7 ClosWait
  Local Socket:   9.67.115.5..1027
  Foreign Socket: 9.37.65.146..21
USER8    000000B8 FinWait2
  Local Socket:   2001:0db8::9:67:115:5..21
  Foreign Socket: 2001:0db8::9:67:115:65..1083
APPM     00000017 UDP
  Local Socket:   ::ffff.0.0.0.0..2051
  Foreign Socket: ::ffff.9.42.103.99..1236
APPV4    00000015 UDP
  Local Socket:   0.0.0.0..2049
  Foreign Socket: 9.42.103.99..1234

SYSLOGD1 0000002C UDP
  Local Socket:   0.0.0.0..529
  Foreign Socket: *..*
```

Report field descriptions:

User Id
See the Client name or User ID information in "Netstat report general concept" on page 324 for a detailed description.

Conn See the Client ID or Connection Number information in "Netstat report general concept" on page 324 for a detailed description.

Local Socket
See the Local Socket information in “Netstat report general concept” on page 324 for a detailed description.

Foreign Socket
See the Foreign Socket information in “Netstat report general concept” on page 324 for a detailed description.

State See the TCP connection status and UDP socket status information in “Netstat report general concept” on page 324 for a detailed description.

Application Data
The application data that makes it easy for users to locate and display the connections that are used by the application. The beginning of the application data identifies the format of the application data area. For z/OS Communications Server applications, see application data information in the z/OS Communications Server: IP Programmer's Guide and Reference for a description of the format, content, and meaning of the data supplied by the application. For other applications, see the documentation that is supplied by the application. The data is displayed in character format if application data is present. Non-printable characters, if any, are displayed as dots.

Netstat ARp/-R report

Queries the ARP cache information. In addition to ARP cache entries for physical devices, when applicable, ARP cache entries for all configured static and dynamic VIPAs are displayed as potential ARP targets, even when they might not be used.

Tip: This report can also display all IPv4 addresses on the HiperSockets internal LAN to which the stack has a route over this interface.

Guideline: For HiperSockets interfaces, the stack requests this data from the appropriate device. If a device does not return this data in a timely fashion, then Netstat will not display data for that interface.

TSO syntax:

```
►►──NETSTAT ARp──┤ Modifier ├──┤ Target ├──┤ Output ├────────────────────────►◄
```

Modifier:

```
►►──┬─netAddress─┬──────────────────────────────────────────────────────────►◄
    └─ALL────────┘
```

netAddress
Queries the ARP cache for a given address.

ALL Queries all ARP cache entries. In addition to ARP cache entries for physical devices when applicable, ARP cache entries for all configured static and dynamic VIPAs are displayed as potential ARP targets, even when they might not be used.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

z/OS UNIX syntax:

```
►►──netstat -R──┤ Modifier ├──┤ Target ├──┤ Output ├──────────────────────────►◄
```

Modifier:

```
►►──┬─netAddress─┬──────────────────────────────────────────────────────────►◄
    └─ALL────────┘
```

netAddress
Queries the ARP cache for a given address.

ALL Queries all ARP cache entries. In addition to ARP cache entries for physical devices when applicable, ARP cache entries for all configured static and dynamic VIPAs are displayed as potential ARP targets even when they might not be used.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See “The Netstat command target” on page 316 for more information about the **-p** parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT ARP 201.2.10.32
Queries the ARP cache for 201.2.10.32 in the default TCP/IP stack.
NETSTAT ARP ALL TCP TCPCS6
Queries all ARP cache entries in TCPCS6 stack.
```

From UNIX shell environment:

```
   netstat -R 201.2.10.32
   netstat -R ALL -p tcpcs6
```

Report examples:
The following examples are generated using the TSO NETSTAT command. The z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT ARP ALL
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           12:48:54
Querying ARP cache for address 201.2.10.32
Interface: SZ_TR1            IBMTR: 08005A0D97A2
Route info: 0270

Querying ARP cache for address 201.2.10.31
Interface: SZ_TR1            IBMTR: 08005A0D97A2
Route info: 0270

Querying ARP cache for address 9.67.128.1
Interface: IQDIOLNKC0010203  IPAQIDIO

Querying ARP cache for address 9.67.1.8
Interface: OSA90LINK1        NSAP: 39999999999999999999ABCDEFABCD1234567890

Querying ARP cache for address 172.16.0.1
Interface: OSXCAINT1         ETHERNET: 420003AA0E56

Querying ARP cache for address 172.16.0.2
Interface: OSXCAINT1         ETHERNET: 420003AA0E5A

Querying ARP cache for address 172.16.0.1
Interface: EZAIQXCA          IPAQIQDX: 820005AA0E0E  OSX: OSXCAINT1

Querying ARP cache for address 172.16.0.2
Interface: EZAIQXCA          IPAQIQDX: 820006AA0E22  OSX: OSXCAINT1
```

```
NETSTAT ARP 201.2.10.32
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           12:48:54
Querying ARP cache for address 201.2.10.32
Interface: SZ_TR1            IBMTR: 08005A0D97A2
Route info: 0270
```

Tip: This report does not reflect information for certain devices that support ARP offload. The information provided differs depending on the type of device. See the z/OS Communications Server: IP Configuration Reference or the z/OS Communications Server: SNA Network Implementation Guide for more information.

Report field descriptions:

IP address
The IP address from the ARP cache.

Interface
The interface name.

Interface Type
The interface type.

MAC address
The MAC address associated with the IP address. This field is not displayed for HiperSockets links.

OSX
For HiperSockets interfaces that use the Internal Queued Direct I/O extensions function (IQDX), this field indicates the associated OSX interface.

Route info
The Token Ring Routing Information Field (RIF). See the RIF portion of RFC 1042 for detailed information about this field. This field is displayed only for Token Ring links.

Netstat BYTEinfo/-b report

Displays byte-count information for each active TCP connection and UDP socket.

TSO syntax:

```
►►──NETSTAT  BYTEinfo──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────►◄
```

Modifier:

```
►►──IDLETIME──────────────────────────────────────────────────────────────►◄
```

IDLETIME

Displays byte-count information plus the idle time for each TCP connection and UDP socket.

Idle time is displayed in the following format:

hours:minutes:seconds.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
                  ┌───────────────┐
                  ▼               │
►►──┬─CLIent────────clientname────┴──────┬──────────────────────────────►◄
    ├─HOSTName──hostname─────────────────┤
    │             ┌────────────────────┐ │
    │             ▼                    │ │
    ├─IPAddr────────┬─ipaddr───────────┬─┴─┤
    │               ├─ipaddr/prefixLen─┤   │
    │               └─ipaddr/subnetmask┘   │
    └─NOTN3270─────────────────────────────┘
```

z/OS UNIX syntax:

```
►►──netstat  -b──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────►◄
```

Modifier:

```
►►──IDLETIME──────────────────────────────────────────────────────────────►◄
```

IDLETIME

Displays the byte-count information plus the idle time for each TCP connection and UDP socket.

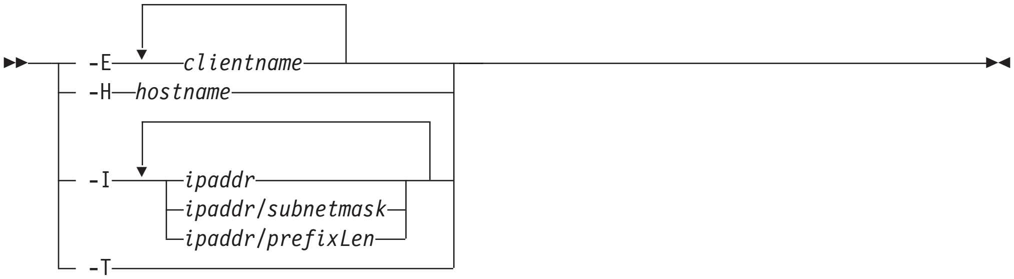
The idle time is displayed in the following format:

hours:minutes:seconds

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:



Filter description:

CLIent/-E clientname
Filter the output of the BYTEinfo/**-b** report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be up to eight characters in length.

HOSTName/-H hostname
Filter the output of the BYTEinfo/**-b** report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters in length.

Result: At the end of the report, Netstat displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value, depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*
Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the BYTEinfo/**-b** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength

Filter the output of the BYTEinfo/**-b** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the BYTEinfo/**-b** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be either the local or remote IP address.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and will usually provide the same result as its IPv4 address.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

NOTN3270/-T

Filter the output of the BYTEinfo/**-b** report, excluding TN3270 server connections.

The filter value for CLIent/**-E** and IPAddr/**-I** can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/**-I** filter, specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/subnetmask* or *ipaddr/prefixlen* format of IPAddr/**-I** values.

When using the z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -b -I '10.*.0.0'** or **netstat -b -I "10.*.0.0"**.

Command syntax examples:

From TSO environment:

```
NETSTAT BYTEINFO
   Displays the byte-count information about each TCP connection and UDP socket in the
   default TCP/IP stack.
NETSTAT BYTEINFO TCP TCPCS6
   Displays the byte-count information about each TCP connection and UDP socket in
   TCPCS6 stack.
NETSTAT BYTEINFO TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Displays the byte-count information about each TCP connection and UDP socket in
   TCPCS8 stack whose foreign IP addresses match the specified filter IP address values.
```

From UNIX shell environment:

```
   netstat -b
   netstat -b -p tcpcs6
   netstat -b -p tcpcs6 -I 9.43.1.1 9.43.2.2
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT BYTEINFO
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           17:19:18
06/06/2003          MVS TCP/IP Real Time Network Monitor
User Id  B Out      B In       L Port  Foreign Socket          State
-------  -----      ----       ------  --------------          -----
FTPD1    0000000000 0000000000 00021   0.0.0.0..0              Listen
FTPD1    0000001062 0000000480 00021   9.67.115.5..1026        Establsh
FTPD1    0000000200 0000000028 00021   9.27.13.21..3711        Establsh
TCPCS    0000000000 0000000000 00023   0.0.0.0..0              Listen
TCPCS    0000000480 0000001062 00023   9.27.11.182..4886       Establsh
APPV4    0000000200 0000000100 02049   9.42.103.99..1234       UDP
SYSLOGD1 0000000000 0000000000 00514   *..*                    UDP
Connections displayed: 6
```

```
NETSTAT BYTEINFO IDLETIME
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:40:44
06/06/2003          MVS TCP/IP Real Time Network Monitor
User Id  B Out   B In    LPort Foreign Socket         State    IdleTime
-------- ------- ------- ----- ---------------------- -------- --------
FTPD1    0000000 0000000 00021 0.0.0.0..0             Listen   00:03:31
FTPD1    0001062 0000480 00021 9.67.115.5..1026       Establsh 00:03:45
FTPD1    0000200 0000028 00021 9.27.13.21..3711       Establsh 00:03:57
TCPCS    0000000 0000000 00023 0.0.0.0..0             Listen   00:01:02
TCPCS    0000480 0001062 00023 9.27.11.182..4886      Establsh 00:04:00
APPV4    0000200 0000100 02049 9.42.103.99..1234      UDP      00:03:01
SYSLOGD1 0000000 0000000 00514 *..*                   UDP      00:02:13
Connections displayed: 6
```

Guideline: For the NETSTAT BYTEINFO IDLETIME display, the byte outbound (B Out) and byte inbound (B In) counts are in three forms:

nnnnnnn
   Number range 0 – 9 999 999

*nnnnnn***K**
   Number range 10 000 000 – 999 999 499 (K = *nnnnnn* x 1000)

*nnnnnn***M**
: Number range 999 999 500 – 4 294 967 287 (M = *nnnnnn* x 1 000 000)

IPv6 enabled or request for LONG format:

```
NETSTAT BYTEINFO
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           16:49:32
06/06/2003          MVS TCP/IP Real Time Network Monitor
User Id  BytesOut             BytesIn              LPort State
-------  --------             -------              ----- -----
FTPD1    00000000000000000000 00000000000000000000 00021 Listen
  Foreign Socket: ::..0
FTPD1    00000000000000000217 00000000000000000025 00021 Establsh
  Foreign Socket: ::ffff:9.67.115.65..1026
FTPD1    00000000000000000438 00000000000000000061 00021 Establsh
  Foreign Socket: 2001:0db8::9:67:115:65..1027
TCPCS    00000000000000000000 00000000000000000000 00023 Listen
  Foreign Socket: 0.0.0.0..0
TCPCS    00000000000000000480 00000000000000001062 00023 Establsh
  Foreign Socket: 9.27.11.182..4665
USER3    00000000000000000000 00000000000000097865 01079 Establsh
  Foreign Socket: 2001:0db8::9:67:115:65..21
USER6    00000000000000000061 00000000000000000438 01027 Establsh
  Foreign Socket: 9.37.65.146..21
APPV4    00000000000000000200 00000000000000000100 02049 UDP
  Foreign Socket: 9.42.103.99..1234
APPV6    00000000000000000200 00000000000000000100 02050 UDP
  Foreign Socket: 12ab::1..1235

SYSLOGD1 00000000000000000000 00000000000000000000 00529 UDP
  Foreign Socket: *..*
Connections displayed: 8
```

```
NETSTAT BYTEINFO IDLETIME
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           16:49:32
06/06/2003          MVS TCP/IP Real Time Network Monitor
User Id  BytesOut             BytesIn              LPort State
-------  --------             -------              ----- -----
FTPD1    00000000000000000000 00000000000000000000 00021 Listen
  Foreign Socket: ::..0
FTPD1    00000000000000000217 00000000000000000025 00021 Establsh
  Foreign Socket: ::ffff:9.67.115.65..1026
FTPD1    00000000000000000438 00000000000000000061 00021 Establsh
  Foreign Socket: 2001:0db8::9:67:115:65..1027
TCPCS    00000000000000000000 00000000000000000000 00023 Listen
  Foreign Socket: 0.0.0.0..0
TCPCS    00000000000000000480 00000000000000001062 00023 Establsh
  Foreign Socket: 9.27.11.182..4665
USER3    00000000000000000000 00000000000000097865 01079 Establsh
  Foreign Socket: 2001:0db8::9:67:115:65..21
USER6    00000000000000000061 00000000000000000438 01027 Establsh
  Foreign Socket: 9.37.65.146..21
APPV4    000000000000000200 000000000000000100 02049 UDP 00:03:01
  Foreign Socket: 9.42.103.99..1234
APPV6    000000000000000200 000000000000000100 02050 UDP 00:20:02
  Foreign Socket: 12ab::1..1235

SYSLOGD1 00000000000000000000 00000000000000000000 00529 UDP
  Foreign Socket: *..*
Connections displayed: 8
```

Guideline: For the NETSTAT BYTEINFO IDLETIME display, the BytesOut and BytesIn counts are in two forms:

nnnnnnnnnnnnnnnnnn
: Number range 0 – 999 999 999 999 999 999

*nnnnnnnnnnnnnnnnn***K**
Number range 1 000 000 000 000 000 000 – 999 999 999 999 999 999 499 (K = *nnnnnnnnnnnnnnnnn* x 1000)

Report field descriptions:

User Id
See the Client name or User ID information in “Netstat report general concept” on page 324 for a detailed description.

BytesIn / B In
For a TCP entry, the number of bytes of data the stack has received for this TCP connection. This includes both the total number of bytes that the application has received and the total number of bytes in the receive buffer that have not yet been read by the application. For a UDP entry, it is the number of bytes of data the stack has received for this UDP socket. This includes both the total number of bytes that all applications have received for this socket and the total number of bytes in stack buffers that have not yet been read by any application.

Restriction: The TCP/IP stack maintains 64-bit counters for TCP connections and UDP endpoints. However, if you are running an IPv4-only stack, and the Netstat output is in the SHORT format, only the lower 32-bit counter value is displayed. If a large amount of data has been received, the number of bytes can exceed a 32-bit counter so the value displayed will appear to have been reset. Use the FORMAT/-M LONG output option on the Netstat command to cause Netstat to use the LONG format for the output. The LONG format displays the full 64-bit counter value. You can also specify the FORMAT parameter on the IPCONFIG profile statement to set FORMAT LONG as the default value for all Netstat commands.

BytesOut / B Out
For a TCP entry, it is the number of bytes of data the application has sent. This includes all of the data that has been sent to the remote connection and all of the data that has not been sent but is buffered and waiting to be sent by the local stack. For a UDP entry, it is the number of outbound bytes of user data sent from this socket.

Restriction: The TCP/IP stack maintains 64-bit counters for TCP connections and UDP endpoints. However, if you are running an IPv4-only stack, and the Netstat output is in the SHORT format, only the lower 32-bit counter value is displayed. If a large amount of data has been sent, the number of bytes can exceed a 32-bit counter so the value displayed will appear to have been reset. Use the FORMAT/-M LONG output option on the Netstat command to cause Netstat to use the LONG format for the output. The LONG format displays the full 64-bit counter value. You can also specify the FORMAT parameter on the IPCONFIG profile statement to set FORMAT LONG as the default value for all Netstat commands.

LPort See the Local port description in “Netstat report general concept” on page 324 for a detailed description.

Foreign Socket
See the Foreign socket information in “Netstat report general concept” on page 324 for a detailed description.

State See the TCP connection status and UDP socket status information in “Netstat report general concept” on page 324 for a detailed description.

IdleTime

The time interval between the current time and the last time the connection was touched. See Last touched time in "Netstat report general concept" on page 324 for a detailed description of the last touched time.

Netstat CACHinfo/-C report

Displays statistics for TCP listening sockets that are using the Fast Response Cache Accelerator (FRCA). For more information about the FRCA, see the Fast Response Cache Accelerator information in z/OS Communications Server: IP Configuration Guide.

TSO syntax:

```
►►──NETSTAT CACHinfo───┤ Target ├──┤ Output ├─────────────────────────────────►◄
```

Target:

Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

z/OS UNIX syntax:

```
►►──netstat  -C──┤ Target ├──┤ Output ├───────────────────────────────────────►◄
```

Target:

Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT CACHINFO
   Displays information about Fast Response Cache Accelerator statistics for the default
   TCP/IP stack.
NETSTAT CACHINFO TCP TCPCS6
   Displays information about Fast Response Cache Accelerator statistics for the TCPCS6
   stack.
```

From UNIX shell environment:

```
   netstat -C
   netstat -C -p tcpcs6
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT CACHINFO
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          13:38:04
Client: USER34          Listening socket:  0.0.0.0..8080
  CacheType:          Exclusive   ASID:                     0030
  MaxCacheSize:       0000000100  CurrCacheSize:      0000000000
  MaxNumObjects:      0000000010  CurrNumObjects:     0000000000
  NumConns:           0000000000  ConnsProcessed:     0000000000
  ConnsDeferred:      0000000000  ConnsTimedOut:      0000000000
  RequestsProcessed:  0000000000  IncompleteRequests: 0000000000
  NumCacheHits:       0000000000  NumCacheMisses:     0000000000
  NumUnprodCacheHits: 0000000000
Client: USER34          Listening socket:  0.0.0.0..8081
  CacheType:          Shared      ASID:                     0030
  MaxCacheSize:       0000000100  CurrCacheSize:      0000000000
  MaxNumObjects:      0000000010  CurrNumObjects:     0000000000
  NumConns:           0000000000  ConnsProcessed:     0000000000
  ConnsDeferred:      0000000000  ConnsTimedOut:      0000000000
  RequestsProcessed:  0000000000  IncompleteRequests: 0000000000
  NumCacheHits:       0000000000  NumCacheMisses:     0000000000
  NumUnprodCacheHits: 0000000000
```

IPv6 enabled or request for LONG format:

```
NETSTAT CACHINFO
EZD0101I NETSTAT CS V2R1 TCPCS 212
CLIENT: USER34
  LISTENING SOCKET: 0.0.0.0..8080
  CACHETYPE:          SHARED      ASID:                     0036
  MAXCACHESIZE:       0000000100  CURRCACHESIZE:      0000000002
  MAXNUMOBJECTS:      0000000010  CURRNUMOBJECTS:     0000000002
  NUMCONNS:           0000000004  CONNSPROCESSED:     0000000002
  CONNSDEFERRED:      0000000002  CONNSTIMEDOUT:      0000000000
  REQUESTSPROCESSED:  0000000003  INCOMPLETEREQUESTS: 0000000001
  NUMCACHEHITS:       0000000003  NUMCACHEMISSES:     0000000002
  NUMUNPRODCACHEHITS: 0000000000
CLIENT: USER34
  LISTENING SOCKET: 0.0.0.0..8081
  CACHETYPE:          SHARED      ASID:                     0036
  MAXCACHESIZE:       0000000100  CURRCACHESIZE:      0000000002
  MAXNUMOBJECTS:      0000000010  CURRNUMOBJECTS:     0000000002
  NUMCONNS:           0000000000  CONNSPROCESSED:     0000000000
  CONNSDEFERRED:      0000000000  CONNSTIMEDOUT:      0000000000
  REQUESTSPROCESSED:  0000000000  INCOMPLETEREQUESTS: 0000000000
  NUMCACHEHITS:       0000000000  NUMCACHEMISSES:     0000000000
  NUMUNPRODCACHEHITS: 0000000000
CLIENT: USER34
  LISTENING SOCKET: 0.0.0.0..8082
  CACHETYPE:          EXCLUSIVE   ASID:                     0036
  MAXCACHESIZE:       0000000100  CURRCACHESIZE:      0000000001
  MAXNUMOBJECTS:      0000000010  CURRNUMOBJECTS:     0000000001
  NUMCONNS:           0000000002  CONNSPROCESSED:     0000000000
  CONNSDEFERRED:      0000000002  CONNSTIMEDOUT:      0000000000
  REQUESTSPROCESSED:  0000000000  INCOMPLETEREQUESTS: 0000000000
  NUMCACHEHITS:       0000000000  NUMCACHEMISSES:     0000000002
  NUMUNPRODCACHEHITS: 0000000000
```

Report field descriptions:
For each listening socket configured for Cache Accelerator support, the following information is displayed:

Client The user name of the application that bound the listening socket.

Socket
The local IP address and port pair to which the listening socket is bound.

CacheType
The type of FRCA cache that is used by the listening socket. It can be one of the following value:

Shared
The cache can be shared by more than one listening socket in the same address space. All listening sockets in the same address space that use a shared cache can access objects stored in the shared cache. Listening sockets from different address spaces cannot access objects that are stored in the cache for a different address space. The values of the MaxCacheSize, CurrCacheSize, MaxNumObjects, and CurrNumObjects fields are the same for all sockets that share a cache.

Exclusive
The cache can be used only by the listening socket. No other listening socket has access to objects stored in the cache.

ASID The hexadecimal address space identifier for the address space that is making the request to enable FRCA on the listening socket.

MaxCacheSize
The maximum number of 4K pages that can be used for storing cache objects by the Cache Accelerator for the given socket.

CurrCacheSize
The number of 4K pages that are currently being used by the Cache Accelerator for storing cache objects.

MaxNumObjects
The maximum number of cache objects that can be stored by the Cache Accelerator.

CurrNumObjects
The current number of cache objects that are stored by the Cache Accelerator.

NumConns
The number of connections established through a listening socket that have been configured with Cache Accelerator support.

ConnsProcessed
The number of connections that have successfully completed an in-kernel transaction, resulting in a response being transmitted to the client. This counter is incremented at most one time per connection.

Tip: It is possible for a single connection to be processed by the Cache Accelerator for some cache entries and then deferred to the application for additional processing. If this occurs, the connection is included in both the ConnsProcessed and ConnsDeferred counts.

ConnsDeferred
The number of connections that require user-space application processing.

Tip: This counter is not incremented because of the connection timeout expiration, even if the action taken is to defer the connection.

ConnsTimedOut
The number of times the connection timeout timer has expired.

RequestsProcessed
The number of requests that were at least partially processed by the Cache Accelerator. This counter can be incremented multiple times for a single connection.

Tip: It is possible for a single connection to be processed by the Cache Accelerator for some cache objects and then deferred to the application for additional processing. If this occurs, the connection is included in both the RequestsProcessed and RequestsDeferred counts.

IncompleteRequests
The number of times that a request is received from the client where additional data is required to process the request. This counter can be incremented multiple times for a single connection.

NumCacheHits
The number of cache objects that were successfully located and transmitted to clients.

NumCacheMisses
The number of cache objects that were not successfully located and transmitted to clients.

NumUnprodCacheHits
The number of cache entries that were successfully found within the cache but not transmitted to the client.

Netstat CLients/-e report

Displays information about local IPv4 users of TCP/IP services (job names).

TSO syntax:

```
►►──NETSTAT CLients──┤ Target ├──┤ Output ├──┤ (Filter ├────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
                  ┌─────────────┐
                  ▼             │
►►──┬─CLIent────────clientname─┴─┬──────────────────────────────────►◄
    └─NOTN3270───────────────────┘
```

z/OS UNIX syntax:

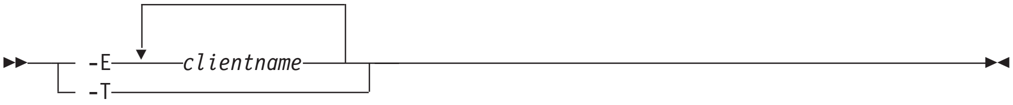
```
►►──netstat  -e──┤ Target ├──┤ Output ├──┤ Filter ├──────────────────────►◄
```

Target:

Provide the report for a specific TCP/IP address space by using the **-p** *tcpname* option. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Filter:



Filter description:

CLIent/-E *clientname*
: Filter the output of the CLients/**-e** report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be up to eight characters in length.

NOTN3270/-T
: Filter the output of the CLients/**-e** report, excluding TN3270 server connections.

The filter value for CLIent/**-E** can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*".

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the client name, new*clnt for the **-E** filter, issue the command as: **netstat -e -E 'new*clnt'** or **netstat -e -E "new*clnt"**.

Command syntax examples:

From TSO environment:

```
NETSTAT CLIENTS
   Display information for each client in the default TCP/IP stack.
NETSTAT CLIENTS TCP TCPCS6
   Display information for each client in TCPCS6 stack.
NETSTAT CLIENTS TCP TCPCS8 (CLIent CSCLNT1 OSGMEM1
   Display information for these clients in TCPCS8 stack whose client name
   match the specified filter client name values.
```

From UNIX shell environment:

```
netstat -e
netstat -e -p tcpcs6
netstat -e -p tcpcs6 -E CSCLNT1 OSGMEM1
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT CLIENTS
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS       12:34:56
Current Clients:

Client: INETD1
Authorization: Autologged
Last Touched:   4:01:17

Client: TCPCS
Authorization: None
Last Touched:   3:14:47
```

Report field descriptions:

Client See Client name or User ID descriptions in “Netstat report general concept” on page 324 for detailed description.

Authorization
The only values that can currently be shown here are *Autologged* and *None*. In earlier versions and releases of TCP/IP for MVS and z/OS, certain types of authorizations for users of TCP/IP services could be configured in the TCP/IP configuration data set. That practice has, over the years, been abandoned and security-related information is now specified in RACF or an equivalent security product.

The following list shows the valid values for this field:

Autologged
This service is being monitored by the TCP/IP autolog function, based on definitions in the AUTOLOG and PORT statements of the TCP/IP profile.

None No special client authorizations.

Last touched time
See the Last touched time information in “Netstat report general concept” on page 324 for a detailed description.

Netstat CONFIG/-f report

Displays TCP/IP configuration information about IP, TCP, UDP, SMF parameters, GLOBALCONFIG profile statement, network monitor, data trace, and autolog settings.

TSO syntax:

```
►►──NETSTAT CONFIG───┤ Target ├──┤ Output ├──────────────────────────────►◄
```

Target:

Provide the report for a specific TCP/IP address space by using the TCp *tcpname* parameter. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

z/OS UNIX syntax:

```
►►──netstat  -f──┤ Target ├──┤ Output ├──────────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using the **-p** *tcpname* option. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT CONFIG
Display the TCP/IP configuration information for the default TCP/IP stack.
NETSTAT CONFIG TCP TCPCS6
Display the TCP/IP configuration information for TCPCS6 stack.
```

From UNIX shell environment:

```
   netstat -f
   netstat -f -p tcpcs6
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT CONFIG
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:37:31
TCP Configuration Table:
DefaultRcvBufSize:  00016384  DefaultSndBufSize: 00016384
DefltMaxRcvBufSize: 00262144  SoMaxConn:         0000001024
MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
RoundTripGain:      0.125     VarianceGain:      0.250
VarianceMultiplier: 2.000     MaxSegLifeTime:    30.000
DefaultKeepALive:   00000120  DelayAck:          Yes
RestrictLowPort:    Yes       SendGarbage:       No
TcpTimeStamp:       Yes       FinWait2Time:      010
TTLS:               No        EphemeralPorts:    1024-65535
SelectiveACK:       Yes       TimeWaitInterval:  30
DefltMaxSndBufSize  262144    RetransmitAttempt: 15
ConnectTimeOut:     0120      ConnectInitIntval: 1000
KeepAliveProbes:    10        KAProbeInterval:   060
Nagle:              No        QueuedRTT:         20
FRRThreshold:       3

UDP Configuration Table:
DefaultRcvBufSize: 00065535  DefaultSndBufSize: 00065535
CheckSum:          Yes       EphemeralPorts:    1024-65535
RestrictLowPort:   Yes       UdpQueueLimit:     No

IP Configuration Table:
Forwarding: Yes    TimeToLive: 00064  RsmTimeOut:  00060
IpSecurity: Yes
ArpTimeout: 01200  MaxRsmSize: 65535  Format:      Short
IgRedirect: Yes    SysplxRout: No     DoubleNop:   No
StopClawEr: No     SourceVipa: Yes
MultiPath:  Conn   PathMtuDsc: No     DevRtryDur:  0000000090
DynamicXCF: Yes
  IpAddr/PrefixLen: 193.9.200.3/28      Metric: 01
  SecClass: 008  SrcVipaInt: IPV4SRCVIPA
QDIOAccel:  No
IQDIORoute: No
TcpStackSrcVipa: 201.1.10.10
ChecksumOffload: Yes             SegOffload: Yes

SMF Parameters:
Type 118:
  TcpInit:      00   TcpTerm:      02   FTPClient:    03
  TN3270Client: 04   TcpIpStats:   05
Type 119:
  TcpInit:      Yes  TcpTerm:      Yes  FTPClient:    Yes
  TcpIpStats:   Yes  IfStats:      Yes  PortStats:    Yes
  Stack:        Yes  UdpTerm:      Yes  TN3270Client: Yes
  IPSecurity:   No   Profile:      Yes  DVIPA:        Yes
  SmcrGrpStats: Yes  SmcrLnkEvent: Yes

Global Configuration Information:
TcpIpStats: Yes  ECSALimit: 2096128K  PoolLimit: 2096128K
MlsChkTerm: No   XCFGRPID:  11        IQDVLANID: 27
SysplexWLMPoll: 060  MaxRecs:   100
ExplicitBindPortRange:  05000-06023   IQDMultiWrite: Yes
AutoIQDX: AllTraffic
WLMPriorityQ: Yes
  IOPri1 0 1
  IOPri2 2
  IOPri3 3 4
  IOPri4 5 6 FWD
Sysplex Monitor:
  TimerSecs: 0060  Recovery: Yes  DelayJoin: No   AutoRejoin: Yes
  MonIntf:   Yes  DynRoute: Yes  Join:      Yes
zIIP:
  IPSecurity: Yes  IQDIOMultiWrite: Yes
SMCR: Yes
  FixedMemory: 100M  TcpKeepMinInt: 00000300
  PFID: 0018  PortNum: 1 MTU: 1024
  PFID: 0019  PortNum: 2 MTU: 1024
```

```
Network Monitor Configuration Information:
PktTrcSrv: Yes  TcpCnnSrv: Yes  MinLifTim: 3  NtaSrv: Yes
SmfSrv: Yes
  IPSecurity: Yes  Profile: Yes  CSSMTP: Yes  CSMAIL: Yes  DVIPA: Yes

Autolog Configuration Information: Wait Time: 0300
ProcName: FTPD      JobName: FTPD
  ParmString:
  DelayStart: Yes
    DVIPA    TTLS
```

IPv6 enabled or request for LONG format:

```
NETSTAT CONFIG
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           19:54:08
TCP Configuration Table:
DefaultRcvBufSize:  00016384  DefaultSndBufSize: 00016384
DefltMaxRcvBufSize: 00262144  SoMaxConn:         0000001024
MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
RoundTripGain:      0.125     VarianceGain:      0.250
VarianceMultiplier: 2.000     MaxSegLifeTime:    30.000
DefaultKeepALive:   00000120  DelayAck:          Yes
RestrictLowPort:    Yes       SendGarbage:       No
TcpTimeStamp:       Yes       FinWait2Time:      010
TTLS:               No        EphemeralPorts:    1024-65535
SelectiveACK:       Yes       TimeWaitInterval:  30
DefltMaxSndBufSize  262144    RetransmitAttempt: 15
ConnectTimeOut:     0120      ConnectInitIntval: 1000
KeepAliveProbes:    10        KAProbeInterval:   060
Nagle:              No        QueuedRTT:         20
FRRThreshold:       3

UDP Configuration Table:
DefaultRcvBufSize: 00065535  DefaultSndBufSize: 00065535
CheckSum:          Yes       EphemeralPorts:    1024-65535
RestrictLowPort:   Yes       UdpQueueLimit:     No

IP Configuration Table:
Forwarding: Yes    TimeToLive: 00064  RsmTimeOut:  00060
IpSecurity: Yes
ArpTimeout: 01200  MaxRsmSize: 65535  Format:      Long
IgRedirect: Yes    SysplxRout: No     DoubleNop:   No
StopClawEr: No     SourceVipa: Yes
MultiPath:  Conn   PathMtuDsc: No     DevRtryDur:  0000000090
DynamicXCF: Yes
  IpAddr/PrefixLen: 193.9.200.3/28      Metric: 01
  SecClass: 008  SrcVipaInt: IPV4SRCVIPA
QDIOAccel:  Yes       QDIOAccelPriority: 2
IQDIORoute: n/a
TcpStackSrcVipa: 201.1.10.10
ChecksumOffload: Yes             SegOffload: Yes

IPv6 Configuration Table:
Forwarding:    Yes  HopLimit:   00255  IgRedirect:  No
SourceVipa:    Yes  MultiPath:  Conn   IcmperrLim:  00003
IgRtrHopLimit: No
IpSecurity: Yes
  OSMSecClass: 255
DynamicXCF: Yes
  IpAddr: 2001:db8::9:67:115:5
  IntfID: 0009:0067:0011:0001
  SrcVipaInt: IPV6SRCVIPA
  SecClass: 008
TcpStackSrcVipa: IPV6STKSRCVIPA
TempAddresses: Yes
  PreferredLifetime: 24   ValidLifetime: 168
ChecksumOffload: Yes             SegOffload: Yes

SMF Parameters:
Type 118:
  TcpInit:      00   TcpTerm:      02   FTPClient:    03
  TN3270Client: 04   TcpIpStats:   05
Type 119:
  TcpInit:      Yes  TcpTerm:      Yes  FTPClient:    Yes
  TcpIpStats:   Yes  IfStats:      Yes  PortStats:    Yes
  Stack:        Yes  UdpTerm:      Yes  TN3270Client: Yes
  IPSecurity:   No   Profile:      Yes  DVIPA:        Yes
  SmcrGrpStats: Yes  SmcrLnkEvent: Yes
```

```
Global Configuration Information:
TcpIpStats: Yes  ECSALimit: 2096128K  PoolLimit: 2096128K
MlsChkTerm: No   XCFGRPID:  11        IQDVLANID: 27
SysplexWLMPoll: 060  MaxRecs:   100
ExplicitBindPortRange:  05000-06023   IQDMultiWrite: Yes
AutoIQDX: AllTraffic
WLMPriorityQ: Yes
  IOPri1 0 1
  IOPri2 2
  IOPri3 3 4
  IOPri4 5 6 FWD
Sysplex Monitor:
  TimerSecs: 0060  Recovery: Yes  DelayJoin: No   AutoRejoin: Yes
  MonIntf:   Yes   DynRoute: Yes  Join:      Yes
zIIP:
  IPSecurity: Yes  IQDIOMultiWrite: Yes
SMCR: Yes
  FixedMemory: 100M  TcpKeepMinInt: 00000300
  PFID: 0018  PortNum: 1 MTU: 1024
  PFID: 0019  PortNum: 2 MTU: 1024

Network Monitor Configuration Information:
PktTrcSrv: Yes  TcpCnnSrv: Yes  MinLifTim: 3  NtaSrv: Yes
SmfSrv:    Yes
  IPSecurity: Yes  Profile: Yes  CSSMTP: Yes  CSMAIL: Yes  DVIPA: Yes

Data Trace Setting:
JobName: *         TrRecCnt: 00000000  Length: FULL
IpAddr:  *                SubNet: *
PortNum: *

Autolog Configuration Information: Wait Time: 0300
ProcName: FTPD      JobName: FTPD
  ParmString:
  DelayStart: Yes
    DVIPA    TTLS
```

Report field descriptions:

- **TCP Configuration Table**

  Display the following configured TCP information that is defined in the TCPCONFIG and SOMAXCONN profile statements. For more information about each field, see the TCPCONFIG or SOMAXCONN profile statement information in z/OS Communications Server: IP Configuration Reference.

  DefaultRcvBufSize

  The TCP receive buffer size that was defined using the TCPRCVBUFRSIZE parameter in the TCPCONFIG statement. The size is between 256 and TCPMAXRCVBUFRSIZE; the default size is 65536 (64 KB). This value is used as the default receive buffer size for those applications that do not explicitly set the buffer size using SETSOCKOPT(). If the TCPRCVBUFRSIZE parameter was not specified in the TCPCONFIG statement, then the default size 65536 (64 KB) is displayed.

  DefaultSndBufSize

  The TCP send buffer size that was defined using the TCPSENDBFRSIZE parameter in the TCPCONFIG statement. The size is between 256 bytes and TCPMAXSENDBUFRSIZE; the default size is 65536 (64 KB). This value is used as the default send buffer size for those applications that do not explicitly set the buffer size using SETSOCKOPT(). If the TCPSENDBFRSIZE parameter was not specified in the TCPCONFIG statement, then the default size 65536 (64 KB) is displayed.

  DefltMaxRcvBufSize

  The TCP maximum receive buffer size that was defined using the TCPMAXRCVBUFRSIZE parameter in the TCPCONFIG statement. The maximum receive buffer size is the maximum value that an application can set as its receive buffer size using SETSOCKOPT(). The minimum acceptable value is the value that is coded on the TCPRCVBUFRSIZE parameter, the maximum size is 2 MB, and the default size is 256 KB. If you do not have large bandwidth interfaces, you can use this parameter to limit the receive buffer size that an application can set. If the TCPMAXRCVBUFRSIZE parameter was not specified in the TCPCONFIG statement, then the default size 262144 (256 KB) is displayed.

SoMaxConn
The maximum number of connection requests that can be queued for any listening socket, as defined by the SOMAXCONN statement. The minimum value is 1, the maximum value is 2147483647, and the default value is 1024.

MaxReTransmitTime
The maximum retransmit interval in seconds. The range is 0 - 999.990. The default value is 120.

Rules:

– If none of the following parameters is specified, this MAXIMUMRETRANSMITTIME parameter is used and the MINIMUMRETRANSMITTIME parameters of the following statements are not used.
  - MAXIMUMRETRANSMITTIME on the BEGINROUTES statement
  - MAXIMUMRETRANSMITTIME on the GATEWAY statement
  - MAXIMUMRETRANSMITTIME on the ROUTETABLE statement
  - Max_Xmit_Time on the OSPF_INTERFACE statement
  - Max_Xmit_Time on the RIP_INTERFACE statement

– The TCPCONFIG version is used if no route parameter has been explicitly specified. If the TCPCONFIG version of maximum retransmit time is used, the MINIMUMRETRANSMITTIME value that is specified on the route parameter is not used, which means the value of the minimum retransmit time is 0.

DefaultKeepAlive
The default keepalive interval that was defined using the INTERVAL parameter in the TCPCONFIG statement. It is the number of minutes that TCP waits after it receives a packet for a connection before it sends a keepalive packet for that connection. The range is 0 - 35791 minutes; the default value is 120. The value 0 disables the keepalive function. If the INTERVAL parameter was not specified in the TCPCONFIG statement, then the default interval 120 is displayed.

DelayAck
Indicates whether the DELAYACKS option is enabled or disabled. The value `Yes` indicates that acknowledgments are delayed when a packet is received (the DELAYACKS parameter was defined in the TCPCONFIG profile statement or is in effect by default); the value `No` indicates that acknowledgments are not delayed when a packet is received (the NODELAYACKS parameter was defined in the TCPCONFIG statement).

RestrictLowPort
Indicates whether ports in the range 1 - 1023 are reserved for users by the PORT and PORTRANGE statements. The value `Yes` indicates that RESTRICTLOWPORTS is in effect (the RESTRICTLOWPORTS parameter

was defined in the TCPCONFIG profile statement); the value No indicates that RESTRICTLOWPORTS is not in effect (the UNRESTRICTLOWPORTS parameter was defined in the TCPCONFIG statement or is in effect by default).

SendGarbage
Indicates whether the keepalive packets sent by TCP contain 1 byte of random data. The value Yes indicates that SENDGARBAGE TRUE is in effect (SENDGARBAGE TRUE was defined in the TCPCONFIG profile statement); the value No indicates that SENDGARBAGE TRUE is not in effect (SENDGARBAGE FALSE was defined in the TCPCONFIG statement or is in effect by default).

TcpTimeStamp
Indicates whether the TCP Timestamp Option is enabled or disabled. The value Yes indicates that TCPTIMESTAMP is in effect (the TCPTIMESTAMP parameter was defined in the TCPCONFIG profile statement or is in effect by default); the value No indicates that TCPTIMESTAMP is not in effect (the NOTCPTIMESTAMP parameter was defined in the TCPCONFIG statement).

FinWait2Time
The FinWait2Time number that was defined using the FINWAIT2TIME parameter in the TCPCONFIG statement. It is the number of seconds a TCP connection should remain in the FINWAIT2 state. The range is 60 - 3600 seconds; the default value is 600 seconds. When this timer expires, it is reset to 75 seconds; when this timer expires a second time, the connection is dropped. If the FINWAIT2TIME parameter was not specified in the TCPCONFIG statement, then the default value 600 is displayed.

TimeWaitInterval
The number of seconds that a connection remains in TIMEWAIT state. The range is 0 - 120. The default value is 60.

Note: For local connections, a TIMEWAITINTERVAL of 50 milliseconds is always used.

DefltMaxSndBufSize
The maximum send buffer size. The range is the value that is specified on TCPSENDBFRSIZE to 2 MB. The default value is 256K.

RetransmitAttempt
The number of times a segment is retransmitted before the connection is aborted. The range is 0 - 15. The default value is 15.

ConnectTimeOut
The total amount of time before the initial connection times out. This value also applies to TCP connections that are established over SMC-R links. The range is 5 - 190 seconds. The default value is 75.

ConnectInitIntval
The initial retransmission interval for the connect(). The range is 100 to 3000 milliseconds (ms). The default value is 3000.

KAProbeInterval
The interval in seconds between keepalive probes. The range is 1 - 75. The default value is 75.

This parameter does not change the initial keepalive timeout interval. It controls the time between the probes that are sent only after the initial keepalive interval has expired.

You can specify setsockopt() TCP_KEEPALIVE to override the parameter.

KeepAliveProbes
The number of keepalive probes to send before the connection is aborted. The range is 1 - 10. The default value is 10.

This parameter does not change the initial keepalive timeout interval. It controls the number of probes that are sent only after the initial keepalive interval has expired.

You can specify setsockopt() TCP_KEEPALIVE to override this parameter.

Nagle Indicates whether the Nagle option is enabled or disabled. The value Yes indicates that packets with less than a full maximum segment size (MSS) of data are buffered unless all data on the send queue has been acknowledged.

QueuedRTT
The threshold at which outbound serialization is engaged. The range is 0 - 50 milliseconds. The default value is 20 milliseconds.

FRRTheshold
The threshold of duplicate ACKs for FRR to engage. The range is 1 - 2048. The default value is 3.

TTLS Indicates whether Application Transparent Transport Layer Security (AT-TLS) is active in the TCP/IP stack. The value `Yes` indicates that AT-TLS is active (the TTLS parameter was specified in the TCPCONFIG profile statement). The value `No` indicates that AT-TLS is not active (the NOTTLS parameter was specified in the TCPCONFIG profile statement or is in effect by default).

EphemeralPorts
The range of ephemeral ports that was defined using the EPHEMERALPORTS parameter in the TCPCONFIG statement or by default. The range specified must be within the range of 1024 to 65535. If the EPHEMERALPORTS parameter was not specified in the TCPCONFIG statement, then the default range 1024 - 65535 is displayed.

SelectiveACK
Indicates whether Selective Acknowledgment (SACK) support is active in the TCP/IP stack. This field can have the following values:

Yes Indicates that SACK options are exchanged with partners when transmitting data. The SELECTIVEACK parameter was specified on the TCPCONFIG profile statement.

No Indicates that SACK options will not be exchanged. The NOSELECTIVEACK parameter was specified on the TCPCONFIG profile statement or is in effect by default.

Note: The values displayed in the MaxReTransmitTime, MinReTransmitTime, RoundTripGain, VarianceGain, VarianceMultiplier, and MaxSegLifeTime fields are actual default values that are assigned by the TCP/IP stack; you cannot configure them externally using the TCPCONFIG profile statement. You can override the MaxReTransmitTime, MinReTransmitTime, RoundTripGain, VarianceGain, VarianceMultiplier values on a per-destination basis using either the BEGINROUTES configuration statement, the old GATEWAY configuration statement, or the configuration file for OMPROUTE.

- **UDP Configuration Table**

  Display the following configured UDP information defined in the UDPCONFIG profile statement. For more information about each UDP parameter, see UDPCONFIG profile statement information in the z/OS Communications Server: IP Configuration Reference.

  DefaultRcvBufSize
  The UDP receive buffer size that was defined using the UDPRCVBUFRSIZE parameter in the UDPCONFIG statement. The size is in the range 1 - 65535; the default size is 65535. If the UDPRCVBUFRSIZE parameter was not specified in the UDPCONFIG statement, then the default size 65535 is displayed.

  DefaultSndBufSize
  The UDP send buffer size that was defined using the UDPSENDBFRSIZE parameter in the UDPCONFIG statement. The size is in the range 1 - 65535; the default size is 65535. If the UDPSENDBFRSIZE parameter was not specified in the UDPCONFIG statement, then the default size 65535 is displayed.

  CheckSum
  Indicates whether UDP does check summing. The value `Yes` indicates that UDP check summing is in effect (the UDPCHKSUM parameter was defined in the UDPCONFIG profile statement or is in effect by default); the value `No` indicates that UDP check summing is not in effect (the NOUDPCHKSUM parameter was defined in the UDPCONFIG statement).

  EphemeralPorts
  The range of ephemeral ports that was defined using the EPHEMERALPORTS parameter in the UDPCONFIG statement or by default. The range specified must be within the range of 1024 to 65535. If the EPHEMERALPORTS parameter was not specified in the UDPCONFIG statement, then the default range 1024 - 65535 is displayed.

  RestrictLowPort
  Indicates whether ports 1 - 1023 are reserved for users by the PORT and PORTRANGE statements. The value `Yes` indicates that ports in the range 1 - 1023 are reserved (the RESTRICTLOWPORTS parameter was defined in the UDPCONFIG profile statement); the value `No` indicates that the ports are not reserved (the UNRESTRICTLOWPORTS parameter was defined in the UDPCONFIG statement or is in effect by default).

  UdpQueueLimit
  Indicates whether UDP should have a queue limit on incoming datagrams. The value `Yes` indicates that there is a UDP queue limit in effect (the UDPQUEUELIMIT parameter was defined in the UDPCONFIG profile statement or is in effect by default); the value `No` indicates that a UDP queue limit is not in effect (the NOUDPQUEUELIMIT parameter was defined in the UDPCONFIG statement).

- **IP Configuration Table**

Displays the following configured IP information defined in the IPCONFIG profile statement. For more information about each IP parameter, see the IPCONFIG profile statement information in the z/OS Communications Server: IP Configuration Reference.

Forwarding
Indicates whether the transfer of data between networks is enabled for this TCP/IP stack. Possible values are:

Pkt Indicates that packets that are received but not destined for this stack are forwarded and use multipath routes if they are available on a per-packet basis (the DATAGRAMFWD FWDMULTIPATH PERPACKET was specified in the IPCONFIG profile statement).

Yes Indicates that packets that are received but not destined for this stack are forwarded but do not use multipath routes even if they are available. (the DATAGRAMFWD NOFWDMULTIPATH was specified in the IPCONFIG profile statement or is in effect by default).

No Indicates that packets that are received but that are not destined for this stack are not forwarded in route to the destination (the NODATAGRAMFWD parameter was specified in the IPCONFIG profile statement).

TimeToLive
The time to live value that was defined using the TTL parameter in the IPCONFIG statement. The time to live value is the number of hops that packets originating from this host can travel before reaching the destination. Valid values are in the range 1 - 255; the default value is 64. If the TTL parameter was not specified in the IPCONFIG statement, then the default value 64 is displayed.

RsmTimeOut
The reassembly timeout value that was defined using the REASSEMBLYTIMEOUT parameter in the IPCONFIG statement. It is the amount of time (in seconds) that is allowed to receive all parts of a fragmented packet before discarding the packets received. Valid values are in the range 1 - 240; the default value is 60. If the REASSEMBLYTIMEOUT parameter was not specified in the IPCONFIG statement, then the default value 60 is displayed.

IpSecurity
Indicates whether the IP filtering and IPSec tunnel support is enabled. The value `Yes` indicates that IP security is in effect (the IPSECURITY parameter was defined on the IPCONFIG profile statement). The value `No` indicates that IP security is not in effect.

ArpTimeout
The ARP timeout value that was defined using the ARPTO parameter in the IPCONFIG statement. It indicates the number of seconds between creation or revalidation and deletion of ARP table entries. Valid values are in the range 60 - 86400; the default value is 1200. If the ARPTO parameter was not specified in the IPCONFIG statement, then the default value 1200 is displayed.

MaxRsmSize
The maximum packet size that can be reassembled. If an IP datagram is

fragmented into smaller packets, the complete reassembled datagram cannot exceed this value. Valid values are in the range 576 - 65535; the default value is 65535.

Restriction: The value that is displayed in the MaxRsmSize field is the actual default value that was assigned by the TCP/IP stack; users cannot configure this value externally using the IPCONFIG profile statement.

Format
The stack-wide command format that was defined using the FORMAT parameter in the IPCONFIG statement or that was assigned by default by TCP/IP stack. This field can have the following values:

SHORT
Indicates that the command report is displayed in the short format (the FORMAT SHORT parameter was specified in the IPCONFIG profile statement).

LONG
Indicates that the command report is displayed in the long format (the FORMAT LONG parameter was specified in the IPCONFIG profile statement).

If the FORMAT parameter was not specified in the IPCONFIG profile statement, then the TCP/IP stack assigned the default format based on whether the stack was IPv6 enabled or not. If the stack is IPv6 enabled, then the format value LONG is assigned by default. If the stack is configured for IPv4-only operation, then the format value SHORT is assigned by default. You can override the stack-wide command format using the Netstat FORMAT/**-M** option.

IgRedirect
Indicates whether TCP/IP is to ignore ICMP Redirect packets. This field can have the following values:

Yes Indicates that IGNOREREDIRECT is in effect. The IGNOREREDIRECT parameter was defined on the IPCONFIG profile statement, OMPROUTE has been started and IPv4 interfaces are configured to OMPROUTE, or intrusion detection services (IDS) policy is in effect to detect and discard ICMP Redirects.

No Indicates that ICMP Redirects are not ignored.

SysplxRout
Indicates whether this TCP/IP host is part of an MVS sysplex domain and should communicate interface changes to the workload manager (WLM). This field can have the following values:

Yes Indicates that SYSPLEXROUTING is in effect (the SYSPLEXROUTING parameter was specified in the IPCONFIG profile statement).

No Indicates that SYSPLEXROUTING is not in effect (the NOSYSPLEXROUTING parameter was specified in the IPCONFIG profile statement or is in effect by default).

DoubleNop
Indicates whether to force channel programs for CLAW devices to have two NOP CCWs to end the channel programs. This field can have the following values:

Yes Indicates that CLAWUSEDOUBLENOP is in effect (the CLAWUSEDOUBLENOP parameter was defined on the IPCONFIG profile statement).

No Indicates that CLAWUSEDOUBLENOP is not in effect.

StopClawEr
Indicates whether to stop channel programs (HALTIO and HALTSIO) when a device error is detected. This field can have the following values:

Yes Indicates that STOPONCLAWERROR is in effect (the STOPONCLAWERROR parameter was specified in the IPCONFIG profile statement).

No Indicates that STOPONCLAWERROR is not in effect.

SourceVipa
Indicates whether the TCP/IP stack uses the corresponding virtual IP address in the HOME list as the source IP address for outbound datagrams that do not have an explicit source address. This field can have the following values:

Yes Indicates that SOURCEVIPA is in effect (the SOURCEVIPA parameter was specified in the IPCONFIG profile statement).

No Indicates that SOURCEVIPA is not in effect (the NOSOURCEVIPA parameter was specified in the IPCONFIG profile statement or is in effect by default).

MultiPath
Indicates whether the multipath routing selection algorithm for outbound IP traffic is enabled for this TCP/IP stack. Possible values are:

Pkt Indicates that outbound traffic uses the multipath routes in a round-robin fashion for each outbound packet (the MULTIPATH PERPACKET parameter was specified in the IPCONFIG profile statement).

Conn Indicates that outbound traffic uses the multipath routes in a round-robin fashion for each outbound connection request (the MULTIPATH PERCONNECTION parameter was specified in the IPCONFIG profile statement).

No Indicates that outbound traffic always uses the first active route in a multipath group (the NOMULTIPATH parameter was specified in the IPCONFIG profile statement or is in effect by default).

PathMtuDsc
Indicates whether TCP/IP is to dynamically discover the PMTU, which is the smallest MTU of all the hops in the path. This field can have the following values:

Yes Indicates that PATHMTUDISCOVERY is in effect (the PATHMTUDISCOVERY parameter was specified in the IPCONFIG profile statement),

No Indicates that PATHMTUDISCOVERY is not in effect (the NOPATHMTUDISCOVERY parameter was specified in the IPCONFIG profile statement or is in effect by default).

DevRtryDur
The retry period duration (in seconds) for a failed device or interface

that was defined using the DEVRETRYDURATION parameter in the IPCONFIG statement. TCP/IP performs reactivation attempts at 30 second intervals during this retry period. The default value is 90 seconds. The value 0 indicates an infinite recovery period; reactivation attempts are performed until the device or interface is either successfully reactivated or manually stopped. The maximum value is 4294967295. If the DEVRETRYDURATION parameter was not specified in the IPCONFIG statement, then the default value 90 is displayed.

DynamicXCF

Indicates whether IPv4 XCF dynamic support is enabled for this TCP/IP stack. This field can have the following values:

Yes Indicates that XCF dynamic support is in effect (the DYNAMICXCF parameter was specified in the IPCONFIG profile statement).

No Indicates that XCF dynamic support is not in effect (the NODYNAMICXCF parameter was specified in the IPCONFIG profile statement or is in effect by default).

When XCF dynamic support is in effect, the following information is displayed:

IpAddr

The IPv4 address that was specified for DYNAMICXCF in the IPCONFIG profile statement.

Subnet

The subnet mask that was specified for DYNAMICXCF in the IPCONFIG profile statement.

Guidelines:

1. If the IpAddr/PrefixLen format was used for DYNAMICXCF in the IPCONFIG profile statement, then it is displayed in the same format in the Netstat report. The PrefixLen is the integer value in the range 1 - 32 that represents the number of left-most significant bits for the address mask.
2. If the IPv6_address/prefix_route_len format was used for DYNAMICXCF in the IPCONFIG6 profile statement, then it is displayed in the same format in the Netstat report. The length of routing prefix is an integer value in the range 1 - 128.

Metric The interface routing metric represents the configured cost_metric value to be used by dynamic routing daemons for routing preferences. It is configured using the cost_metric value in the IPCONFIG DYNAMICXCF statement.

SecClass

Indicates the IP Security security class value that is associated with the dynamic XCF link. Valid values are in the range 1 - 255.

SrcVipaInt

The source VIPA interface name that was defined using the DYNAMICXCF SOURCEVIPAINTERFACE parameter in the IPCONFIG statement. It must be a VIRTUAL interface. This field indicates the value No if the SOURCEVIPAINTERFACE subparameter was not specified for the DYNAMICXCF in the IPCONFIG statement.

QDIOAccel

Indicates whether QDIO Accelerator is enabled for this TCP/IP stack. This field can have the following values:

Yes Indicates that the QDIO Accelerator is enabled (the QDIOACCELERATOR parameter was specified in the IPCONFIG profile statement).

SD only

Indicates that the QDIO Accelerator is enabled (the QDIOACCELERATOR parameter was specified in the IPCONFIG profile statement), but only for Sysplex Distributor traffic and not for routed traffic. This might be the case if IP forwarding is disabled on this stack, or if IP filters or defensive filters require this stack to perform special processing for routed traffic. For more information, see QDIO Accelerator and IP security in z/OS Communications Server: IP Configuration Guide.

No Indicates that the QDIO Accelerator is not enabled (the NOQDIOACCELERATOR parameter was specified in the IPCONFIG profile statement or is in effect by default).

QDIOAccelPriority

Indicates which QDIO outbound priority level should be used if the QDIO Accelerator is routing packets to a QDIO device. If the NOQDIOACCELERATOR parameter was specified in the IPCONFIG profile statement or is in effect by default, then the QDIOAccelPriority field is not displayed.

IQDIORoute

Indicates whether HiperSockets Accelerator is enabled for this TCP/IP stack. This field can have the following values:

Yes Indicates that HiperSockets Accelerator is enabled (the IQDIOROUTING parameter was specified in the IPCONFIG profile statement).

No Indicates that HiperSockets Accelerator is not enabled (the NOIQDIOROUTING parameter was specified in the IPCONFIG profile statement or is in effect by default).

n/a Indicates that HiperSockets Accelerator does not apply because QDIO Accelerator is enabled.

QDIOPriority

Indicates which QDIO outbound priority level should be used if the HiperSockets Accelerator is routing packets to a QDIO device. If the NOIQDIOROUTING parameter was specified in the IPCONFIG profile statement or is in effect by default, then the QDIOPriority field is not displayed. This field is displayed only when the IQDIORoute field value is `Yes`.

TcpStackSrcVipa

The IPv4 address that was defined using the TCPSTACKSOURCEVIPA parameter in the IPCONFIG statement. It must be the source IP address for outbound TCP connections if SOURCEVIPA has been enabled. This field has the value `No` if the TCPSTACKSOURCEVIPA parameter was not specified in the IPCONFIG statement

ChecksumOffload

Indicates whether the IPv4 checksum offload function is enabled or disabled. This field can have the following values:

Yes Indicates that the checksum processing for IPv4 packets is offloaded to OSA-Express interfaces that support the checksum offload function. The CHECKSUMOFFLOAD parameter was specified on the IPCONFIG profile statement or the value was set by default.

No Indicates that the checksum processing is performed by the TCP/IP stack. The NOCHECKSUMOFFLOAD parameter was specified on the IPCONFIG profile statement.

SegOffload

Indicates whether the IPv4 TCP segmentation offload function is enabled or disabled. This field can have the following values:

Yes Indicates that IPv4 TCP segmentation is performed by OSA-Express interfaces that support the segmentation offload function. The SEGMENTATIONOFFLOAD parameter was specified on the IPCONFIG profile statement.

No Indicates that the segmentation is performed by the TCP/IP stack. The NOSEGMENTATIONOFFLOAD parameter was specified on the IPCONFIG profile statement or the value was set by default.

- **IPv6 Configuration Table if the TCP/IP stack is IPv6 enabled**

  Displays the following configured IPv6 information that is defined in the IPCONFIG6 profile statement For more information about each IPv6 IP parameter, see the IPCONFIG6 profile statement information in the z/OS Communications Server: IP Configuration Reference.

Forwarding

Indicates whether the transfer of data between networks is enabled for this TCP/IP stack. Possible values are:

Pkt Indicates that packets that are received but that are not destined for this stack are forwarded and use multipath routes if available on a per-packet basis (the DATAGRAMFWD FWDMULTIPATH PERPACKET was specified in the IPCONFIG6 profile statement).

Yes Indicates that packets that are received but that are not destined for this stack are forwarded but do not use multipath routes even if they are available. (the DATAGRAMFWD NOFWDMULTIPATH was specified in the IPCONFIG6 profile statement or is in effect by default).

No Indicates that packets that are received but that are not destined for this stack are not forwarded in route to the destination (the NODATAGRAMFWD parameter was specified in the IPCONFIG6 profile statement).

HopLimit

The hop limit value that was defined using the HOPLIMIT parameter in the IPCONFIG6 statement. It is the number of hops that a packet that originates at this host can travel in route to the destination. Valid values are in the range 1 - 255; the default value is 255. If the HOPLIMIT parameter was not specified in the IPCONFIG6 statement, then the default value 255 is displayed.

IgRedirect
Indicates whether TCP/IP is to ignore ICMP Redirect packets. This field can have the following values:

Yes Indicates that IGNOREREDIRECT is in effect. The IGNOREREDIRECT parameter was defined on the IPCONFIG6 profile statement, OMPROUTE has been started and IPv6 interfaces are configured to OMPROUTE, or intrusion detection services (IDS) policy is in effect to detect and discard ICMP Redirects.

No Indicates that ICMP Redirects are not ignored.

SourceVipa
Indicates whether to use a virtual IP address that is assigned to the SOURCEVIPAINT interface as the source address for outbound datagrams that do not have an explicit source address. You must specify the SOURCEVIPAINT parameter on the INTERFACE profile statement for each interface where you want the SOURCEVIPA address to take effect. This field can have the following values:

Yes Indicates that SOURCEVIPA is in effect (the SOURCEVIPA parameter was specified in the IPCONFIG6 profile statement).

No Indicates that SOURCEVIPA is not in effect (the NOSOURCEVIPA parameter was specified in the IPCONFIG6 profile statement or is in effect by default).

MultiPath
Indicates whether the multipath routing selection algorithm for outbound IP traffic is enabled for this TCP/IP stack. Possible values are:

Pkt Indicates that outbound traffic uses the multipath routes in a round-robin fashion for each outbound packet (the MULTIPATH PERPACKET parameter was specified in the IPCONFIG6 profile statement).

Conn Indicates that outbound traffic uses the multipath routes in a round-robin fashion for each outbound connection request (the MULTIPATH PERCONNECTION parameter was specified in the IPCONFIG6 profile statement).

No Indicates that outbound traffic always uses the first active route in a multipath group (the NOMULTIPATH parameter was specified in the IPCONFIG6 profile statement is in effect by default).

IcmperrLim
The ICMP error limit value that was defined using the ICMPERRORLIMIT parameter in the IPCONFIG6 statement. It controls the rate at which ICMP error messages can be sent to a particular IPv6 destination address. The number displayed is the number of messages per second. Valid values are in the range 1 - 20; the default value is 3. If the ICMPERRORLIMIT parameter was not specified in the IPCONFIG6 statement, then the default value 3 is displayed.

IgRtrHopLimit
Indicates whether the TCP/IP stack ignores a hop limit value that is received from a router in a router advertisement. This field can have the following values:

Yes Indicates that IGNOREROUTERHOPLIMIT is in effect (the IGNOREROUTERHOPLIMIT parameter was defined on the IPCONFIG6 profile statement).

No Indicates that IGNOREROUTERHOPLIMIT is not in effect (the NOIGNOREROUTERHOPLIMIT parameter was defined on the IPCONFIG6 profile statement or is in effect by default).

IpSecurity
Indicates whether the IP filtering and IPSec tunnel support is enabled.

Yes Indicates that IP security is in effect (the IPSECURITY parameter was defined on the IPCONFIG6 profile statement). When IP security is in effect, the following information is displayed:

OSMSecClass
Indicates the IP Security security class value that is associated with the OSM interfaces. Valid values are in the range 1 - 255.

No Indicates that IP security is not in effect.

DynamicXCF
Indicates whether IPv6 XCF dynamic support is enabled for this TCP/IP stack. This field can have the following values:

Yes Indicates that XCF dynamic support is in effect (the DYNAMICXCF parameter was specified in the IPCONFIG6 profile statement).

No Indicates that XCF dynamic support is not in effect (the NODYNAMICXCF parameter was specified in the IPCONFIG6 profile statement or is in effect by default).

When XCF dynamic support is in effect, the following information is displayed:

IpAddr
The IPv6 address that was specified for DYNAMICXCF in the IPCONFIG6 profile statement.

Tip: If the IpAddr/PrefixRouteLen format was used for DYNAMICXCF in the IPCONFIG6 profile statement, then it is displayed in the same format in the Netstat report. The PrefixRouteLen is the integer value in the range 1 - 128.

IntfId The 64-bit interface identifier in colon-hexadecimal format that was specified using INTFID subparameter for DYNAMICXCF in the IPCONFIG6 profile statement. If the INTFID subparameter was not specified, then this field is `not` displayed.

SrcVipaInt
The source VIPA interface name that was defined using the DYNAMICXCF SOURCEVIPAINTERFACE parameter in the IPCONFIG6 statement. It must be a VIRTUAL6 interface. This field indicates the value `No` if the SOURCEVIPAINTERFACE subparameter was not specified for the DYNAMICXCF in the IPCONFIG6 statement.

SecClass
Indicates the IP Security security class value that is associated with the IPv6 dynamic XCF interfaces. Valid values are in the range 1 - 255.

TcpStackSrcVipa

The IPv6 interface name that was defined using the TCPSTACKSOURCEVIPA parameter in the IPCONFIG6 statement. It must be the source interface for outbound TCP connections if SOURCEVIPA has been enabled. This field indicates the value `No` if the TCPSTACKSOURCEVIPA parameter was not specified in the IPCONFIG6 statement

TempAddresses

Indicates whether the TCP/IP stack generates IPv6 temporary addresses for IPv6 interfaces for which stateless address autoconfiguration is enabled. This field can have the following values:

Yes Indicates that this behavior is enabled (the TEMPADDRS parameter was defined on the IPCONFIG6 profile statement).

No Indicates that this behavior is not enabled (the NOTEMPADDRS parameter was defined on the IPCONFIG6 profile statement or is in effect by default).

When TEMPADDRS support is in effect, the following information is displayed:

PreferredLifetime

The preferred lifetime for IPv6 temporary addresses, which was defined using the PREFLIFETIME parameter in the IPCONFIG6 statement.

At the expiration of the preferred lifetime, a new temporary address is generated and the existing address is deprecated. The number that is displayed is the preferred lifetime, in hours. Valid values are in the range of 1 - 720 hours (30 days). The default value is 24 hours.

ValidLifetime

The valid lifetime for IPv6 temporary addresses that was defined using the VALIDLIFETIME parameter in the IPCONFIG6 statement.

When the valid lifetime expires, the temporary address is deleted. The number displayed is the valid lifetime in hours. Valid values are in the range 2 - 2160 hours (90 days). The default value is 7 times the preferred lifetime value, with a maximum of 90 days.

ChecksumOffload

Indicates whether the IPv6 checksum offload function is enabled or disabled. This field can have the following values:

Yes Indicates that the checksum processing for IPv6 packets is offloaded to OSA-Express interfaces that support the checksum offload function. The CHECKSUMOFFLOAD parameter was specified on the IPCONFIG6 profile statement or the value was set by default.

No Indicates that the checksum processing is performed by the TCP/IP stack. The NOCHECKSUMOFFLOAD parameter was specified on the IPCONFIG6 profile statement.

SegOffload

Indicates whether the IPv6 TCP segmentation offload function is enabled or disabled. This field can have the following values:

Yes Indicates that the IPv6 TCP segmentation is offloaded to OSA-Express interfaces that support the segmentation offload function. The SEGMENTATIONOFFLOAD parameter was specified on the IPCONFIG6 profile statement.

No Indicates that the segmentation is performed by the TCP/IP stack. The NOSEGMENTATIONOFFLOAD parameter was specified on the IPCONFIG6 profile statement or the value was set by default.

- **SMF parameters**

  Display the following configured SMF information defined in the SMFCONFIG profile statement. For more information about each SMF parameter, see SMFCONFIG profile statement information in the z/OS Communications Server: IP Configuration Reference.

Type 118

TcpInit

Indicates whether SMF subtype 1 records are created when TCP connections are established. A value of the subtype indicates that TYPE118 TCPINIT is in effect (the TCPINIT or TYPE118 TCPINIT was specified on the SMFCONFIG profile statement or a nonzero value of inittype was specified on the SMFPARMS profile statement).

The value 0 indicates that TYPE118 TCPINIT is not in effect (the NOTCPINIT or TYPE118 NOTCPINIT was specified in the SMFCONFIG profile statement (or is in effect by default), or zero value of inittype was specified on the SMFPARMS profile statement).

TcpTerm

Indicates whether SMF subtype 2 records are created when TCP connections are established. A value of the subtype indicates that TYPE118 TCPTERM is in effect (the TCPTERM or TYPE118 TCPTERM was specified on the profile SMFCONFIG statement or a non zero value of termtype was specified on the SMFPARMS profile statement).

The value 0 indicates that TYPE118 TCPTERM is not in effect (the NOTCPTERM or TYPE118 NOTCPTERM was specified in the SMFCONFIG profile statement (or is in effect by default), or zero value of termtype was specified on the SMFPARMS profile statement).

FTPClient

Indicates whether SMF subtype 3 records are created when TCP connections are established. A value of the subtype indicates that TYPE118 FTPCLIENT is in effect (the FTPCLIENT or TYPE118 FTPCLIENT was specified on the SMFCONFIG profile statement or a non zero value of clienttype was specified on the SMFPARMS profile statement).

The value 0 indicates that TYPE118 FTPCLIENT is not in effect (the NOFTPCLIENT or TYPE118 NOFTPCLIENT was specified in the SMFCONFIG profile statement (or is in effect by default), or zero value of clienttype was specified on the SMFPARMS profile statement).

TN3270Client

Indicates whether SMF subtype 4 records are created when TCP connections are established. A value of the subtype indicates TYPE118 TN3270CLIENT is in effect (the TN3270CLIENT or TYPE118 TN3270CLIENT was specified on the SMFCONFIG profile statement or a non zero value of clienttype was specified on the SMFPARMS profile statement).

The value 0 indicates that TYPE118 TN3270CLIENT is not in effect (the NOTN3270CLIENT or TYPE118 NOTN3270CLIENT was specified in the SMFCONFIG profile statement (or is in effect by default), or zero value of clienttype was specified on the SMFPARMS profile statement).

TcpIpStates

Indicates whether SMF subtype 5 records are created when TCP connections are established. A value of the subtype indicates that TYPE118 TCPIPSTATISTICS is in effect (the TCPIPSTATISTICS or TYPE118 TCPIPSTATISTICS was specified on the SMFCONFIG statement).

The value 0 indicates that TYPE118 TCPIPSTATISTICS is not in effect (the NOTCPIPSTATISTICS or TYPE118 NOTCPIPSTATISTICS was specified in the SMFCONFIG profile statement or is in effect by default).

Type 119

TcpInit

Indicates whether SMF records of subtype 1 are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 TCPINIT is in effect (the TYPE119 TCPINIT was specified on the SMFCONFIG statement).

No Indicates that TYPE119 TCPINIT is not in effect (the TYPE119 NOTCPINIT was specified in the SMFCONFIG profile statement or is in effect by default).

TcpTerm

Indicates whether SMF subtype 2 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 TCPTERM is in effect (the TYPE119 TCPTERM was specified on the SMFCONFIG statement).

No Indicates that TYPE119 TCPTERM is not in effect (the TYPE119 NOTCPTERM was specified in the SMFCONFIG profile statement or is in effect by default).

FTPClient

Indicates whether SMF subtype 3 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 FTPCLIENT is in effect (the TYPE119 FTPCLIENT was specified on the SMFCONFIG statement).

No Indicates that TYPE119 FTPCLIENT is not in effect (the TYPE119 NOFTPCLIENT was specified in the SMFCONFIG profile statement or is in effect by default).

TcpIpStats
Indicates whether SMF subtype 5 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 TCPIPSTATISTICS is in effect (the TYPE119 TCPIPSTATISTICS was specified on the SMFCONFIG statement).

No Indicates that TYPE119 TCPIPSTATISTICS is not in effect (the TYPE119 NOTCPIPSTATISTICS was specified in the SMFCONFIG profile statement or is in effect by default).

IfStats Indicates whether SMF subtype 6 and subtype 44 records are created. This field can have the following values:

Yes Indicates that TYPE119 IFSTATISTICS is in effect (the TYPE119 IFSTATISTICS was specified on the SMFCONFIG statement).

No Indicates that TYPE119 IFSTATISTICS is not in effect (the TYPE119 NOIFSTATISTICS was specified in the SMFCONFIG profile statement or is in effect by default).

PortStats
Indicates whether SMF subtype 7 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 PORTSTATISTICS is in effect (the TYPE119 PORTSTATISTICS was specified on the SMFCONFIG statement).

No Indicates that TYPE119 PORTSTATISTICS is not in effect (the TYPE119 NOPORTSTATISTICS was specified in the SMFCONFIG profile statement or is in effect by default).

Stack Indicates whether SMF subtype 8 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 TCPSTACK is in effect (the TYPE119 TCPSTACK was specified on the SMFCONFIG statement).

No Indicates that TYPE119 TCPSTACK is not in effect (the TYPE119 NOTCPSTACK was specified in the SMFCONFIG profile statement or is in effect by default).

UdpTerm
Indicates whether SMF subtype 10 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 UDPTERM is in effect (the TYPE119 UDPTERM was specified on the SMFCONFIG statement).

No Indicates that TYPE119 UDPTERM is not in effect (the TYPE119 NOUDPTERM was specified in the SMFCONFIG profile statement or is in effect by default).

TN3270Client

Indicates whether SMF subtype 22 and 23 records are created when TCP connections are established. This field can have the following values:

Yes Indicates that TYPE119 TN3270CLIENT is in effect (the TYPE119 TN3270CLIENT was specified on the SMFCONFIG statement).

No Indicates that TYPE119 TN3270CLIENT is not in effect (the TYPE119 NOTN3270CLIENT was specified in the SMFCONFIG profile statement or is in effect by default).

IPSecurity

Indicates whether SMF records of subtypes 77, 78, 79, and 80 are created when dynamic tunnels are removed and when manual tunnels are activated and deactivated. This field can have the following values:

Yes Indicates that TYPE119 IPSECURITY is in effect (the TYPE119 IPSECURITY was specified on the SMFCONFIG statement).

No Indicates that TYPE119 IPSECURITY is not in effect (the TYPE119 NOIPSECURITY was specified or is in effect by default in the SMFCONFIG profile statement).

Profile

Indicates whether SMF subtype 4 event records are created when the TCP/IP stack is initialized or when a profile change occurs. This record provides TCP/IP stack profile information. This field can have the following values:

Yes Indicates that this behavior is enabled (the TYPE119 PROFILE parameter was specified on the SMFCONFIG statement).

No Indicates that this behavior is not enabled (the TYPE119 NOPROFILE parameter was specified on the SMFCONFIG statement or is in effect by default).

DVIPA

Indicates whether SMF subtypes 32, 33, 34, 35, 36, and 37 event records are created for sysplex events. These records provide information about changes to dynamic virtual IP addresses (DVIPAs), DVIPA targets, and DVIPA target servers. This field can have the following values:

Yes Indicates that this behavior is enabled (the TYPE119 DVIPA parameter was specified on the SMFCONFIG statement).

No Indicates that this behavior is not enabled (the TYPE119

NODVIPA parameter was specified on the SMFCONFIG statement or is in effect by default).

SmcrGrpStats

Indicates whether SMF subtype 41 records are created. These records are SMC-R link group statistics records. The records collect information about Shared Memory Communications over Remote Direct Memory Access (SMC-R) link groups and the SMC-R links within each group. This field can have the following values:

Yes Indicates that this behavior is enabled. The TYPE119 SMCRGROUPSTATISTICS parameter was specified on the SMFCONFIG statement.

No Indicates that this behavior is not enabled. The TYPE119 NOSMCRGROUPSTATISTICS parameter was specified on the SMFCONFIG statement or is in effect by default.

SmcrLnkEvent

Indicates whether SMF subtype 42 and 43 records are created. The SMF records of subtype 42 are created when SMC-R links are started, and the SMF records of subtype 43 are created when SMC-R links are ended. This field can have the following values:

Yes Indicates that this behavior is enabled. The TYPE119 SMCRLINKEVENT parameter was specified on the SMFCONFIG statement.

No Indicates that this behavior is not enabled. The TYPE119 NOSMCRLINKEVENT parameter was specified on the SMFCONFIG statement or is in effect by default.

Note: The TCPIP statistics field under SMF Parameters displays the subtype value used when creating the SMF type 118 record (if the value is nonzero). The TCPIP statistics field under Global Configuration Information indicates whether the TCP/IP stack will write statistics messages to the TCP/IP job log when TCP/IP is terminated. For the Type 119 fields, the subtype cannot be changed and the setting indicates if the record is requested (Yes) or not (No).

- **Global Configuration Information**

  Display the following global configured information defined in the GLOBALCONFIG profile statement. For more information about each global parameter, see GLOBALCONFIG profile statement information in the z/OS Communications Server: IP Configuration Reference.

  TcpIpStats

  Indicates whether the several TCP/IP counter values are to be written to the output data set designated by the CFGPRINT JCL statement. The value Yes indicates that TCPIPSTATISTICS is in effect (the TCPIPSTATISTICS parameter was specified in the GLOBALCONFIG profile statement). The value No indicates that TCPIPSTATISTICS is not in effect (the NOTCPIPSTATISTICS parameter was specified in the GLOBALCONFIG profile statement or is in effect by default).

  Tip: The TCPIPSTATS field that is shown under the SMF PARAMETERS section of the Netstat CONFIG/**-f** output reflects the TcpIpStatistics value or NoTcpIpStatistics value that is specified on the SMFCONFIG statement in the TCP/IP Profile or Obeyfile. The TCPIPSTATS field that is shown under the GLOBAL CONFIGURATION section of the Netstat

CONFIG/**-f** output reflects the value from the GLOBALCONFIG statement in the TCP/IP Profile or Obeyfile.

ECSALimit

The maximum amount of extended common service area (ECSA) that was defined using the ECSALIMIT parameter in the GLOBALCONFIG statement. This limit can be expressed as a number followed by the letter K (which represents 1024 bytes), or a number followed by the letter M (which represents 1048576 bytes). If the K suffix is used, then the value displayed must be in the range 10240K - 2096128K inclusive, or 0K. If the M suffix is used, the value displayed must be in the range 10M - 2047M inclusive, or 0K. If the ECSALIMIT parameter was not specified in the GLOBALCONFIG statement, then the default value 0K is displayed (which means no limit).

PoolLimit

The maximum amount of authorized private storage that was defined using the POOLLIMIT parameter in the GLOBALCONFIG statement. This limit can be expressed as a number followed by the letter K (which represents 1024 bytes), or a number followed by the letter M (which represents 1048576 bytes). If the K suffix is used, then the value displayed must be in the range 10240K to 2096128K inclusive, or 0K. If the M suffix is used, value is displayed must be in the range 10M - 2047M inclusive, or 0K. If the POOLLIMIT parameter was not specified in the GLOBALCONFIG statement, then the default value 0K is displayed (which means no limit).

MlsChkTerm

Indicates whether the stack should be terminated when inconsistent configuration information is discovered in a multilevel-secure environment. The value `Yes` indicates that MLSCHKTERMINATE is in effect (the MLSCHKTERMINATE parameter was specified in the GLOBALCONFIG profile statement). The value `No` indicates that MLSCHKTERMINATE is not in effect (the NOMLSCHKTERMINATE parameter was specified in the GLOBALCONFIG profile statement or is in effect by default).

XCFGRPID

Displays the TCP 2-digit XCF group name suffix. The two digits displayed are used to generate the XCF group that the TCP/IP stack has joined. The group name is EZBT*vvtt*, where *vv* is the VTAM XCF group ID suffix (specified as a VTAM start option) and *tt* is the displayed XCFGRPID value. If no VTAM XCF group ID suffix was specified, the group name is EZBTCP*tt*. You can use the D TCPIP,,SYSPLEX,GROUP command to display the group name that the TCP/IP stack has joined.

These digits are also used as a suffix for the EZBDVIPA and EZBEPORT structure names in the form EZBDVIPA*vvtt* and EZBEPORT*vvtt*. If no VTAM XCF group ID suffix was specified, the structure names are EZBDVIPA01*tt* and EZBEPORT01*tt*. If no XCFGRPID value was specified on the GLOBALCONFIG statement in the TCP/IP profile, then no value is displayed for XCFGRPID field in the Netstat output.

IQDVLANID

Displays the TCP/IP VLAN ID that is to be used when a HiperSockets link or interface is generated for dynamic XCF connectivity between stacks on the same CPC. The VLAN ID provides connectivity separation between TCP/IP stacks using HiperSockets for dynamic XCF when subplexing is being used (when XCFGRPID was specified on the

GLOBALCONFIG statement). TCP/IP stacks with the same XCFGRPID value (stacks in the same subplex) should specify the same IQDVLANID value if the stacks are in the same CPC and use the same CHPID value. TCP/IP stacks with different XCFGRPID values should specify different IQDVLANID values if the stacks are in the same CPC and use the same CHPID value. If no IQDVLANID value was specified on the GLOBALCONFIG statement in the TCP/IP profile, then the value 0 (no value) is displayed for the IQDVLANID field in the Netstat output.

SysplexWLMPoll

The rate, in seconds, at which the sysplex distributor and its target servers poll WLM for new weight recommendations. A shorter rate indicates a quicker response; however, shorter rates might result in unneeded queries.

MaxRecs

The maximum number of records that are displayed by the DISPLAY TCPIP,,NETSTAT operator command, if the MAX parameter is not specified on that command. The maximum number of records is specified on the MAXRECS parameter of the GLOBALCONFIG profile statement. An asterisk (*) indicates that all records are displayed.

ExplicitBindPortRange

The range of ephemeral ports that is assigned uniquely across the sysplex when an explicit bind() is issued using INADDR_ANY or the unspecified IPv6 address (in6addr_any) and when the specified port is 0.

Tip: This range is the range that was configured on this stack. It might not be the actual range that is in use throughout the sysplex at this time, because another stack that was started later with a different explicit bind port range configured (or with a VARY OBEYFILE command specifying a file with a different EXPLICITBINDPORTRANGE value) can override the range that is configured by this stack. Use the Display TCPIP,,SYSPLEX,PORTS command to display the currently active port range.

AutoIQDX

Indicates whether dynamic Internal Queued Direct I/O extensions function (IQDX) interfaces are used for connectivity to the intraensemble data network (IEDN). This field can have the following values:

No
Indicates that access to the IEDN using HiperSockets (IQD CHPIDs) with the IQDX is disabled. The NOAUTOIQDX parameter was specified on the GLOBALCONFIG statement.

AllTraffic
Indicates that IQDX interfaces are used for all eligible outbound traffic to the IEDN. The AUTOIQDX ALLTRAFFIC parameter was specified on the GLOBALCONFIG statement. This value is the default value for the AutoIQDX field.

NoLargeData
Indicates that IQDX interfaces are used for all eligible outbound traffic to the IEDN, except for large outbound TCP protocol traffic. The AUTOIQDX NOLARGEDATA parameter was specified on the GLOBALCONFIG statement. Large outbound TCP traffic is sent to the IEDN by using OSX OSA-Express interfaces.

IQDMultiWrite

Indicates whether all HiperSockets interfaces are configured to move multiple output data buffers using a single write operation. You must stop and restart the interface for a change in this value to take effect for an active HiperSockets interface. This field can have the following values:

Yes Indicates that the HiperSockets interfaces are configured to use HiperSockets multiple write support when this function is supported by the IBM System z environment (the IQDMULTIWRITE parameter was specified on the GLOBALCONFIG profile statement).

No Indicates that the HiperSockets interfaces are not configured to use HiperSockets multiple write support (the NOIQDMULTIWRITE parameter was specified on the GLOBALCONFIG profile statement or the value was set by default).

WLMPriorityQ

Indicates whether OSA-Express QDIO write priority values are being assigned to outbound OSA-Express packets that are associated with Workload Manager (WLM) service classes, and to forwarded packets that are not being accelerated. The displayed priorities are applied only when the IPv4 type of service (ToS) byte or the IPv6 traffic class value in the IP header is 0 and the packet is sent from an OSA-Express device that is in QDIO mode. This field can have the following values:

Yes Indicates that QDIO write priority values are assigned to outbound OSA-Express packets that are associated with Workload Manager (WLM) service classes, and to forwarded packets that are not being accelerated (the WLMPRIORITYQ parameter was specified on the GLOBALCONFIG profile statement). When the WLMPriorityQ field has the value `Yes`, the following information is displayed:

IOPRIn control_values

Indicates which QDIO priority value is assigned to each control value. The QDIO priority values are in the range of 1 - 4. These QDIO priority values are displayed as the identifiers `IOPRI1`, `IOPRI2`, `IOPRI3`, and `IOPRI4`. The values that follow the identifiers are the control values. The control values represent Workload Manager service classes and forwarded packets. Most of the control values correlate directly to Workload Manager service class importance levels. See the WLMPRIORITYQ parameter in the GLOBALCONFIG profile statement information in z/OS Communications Server: IP Configuration Reference for more details about the control values. If no control value was specified for a specific QDIO priority value, then the identifier for that QDIO priority value is not displayed.

No Indicates that QDIO write priority values are not assigned to outbound OSA-Express packets that are associated with Workload Manager (WLM) service classes or to forwarded packets that are not accelerated (the NOWLMPRIORITYQ parameter was specified on the GLOBALCONFIG profile statement or is in effect by default).

Sysplex Monitor

Displays the parameter values for the Sysplex Problem Detection and Recovery function.

TimerSecs

Displays the timer value (in seconds) that is used to determine how soon the sysplex monitor timer reacts to problems with needed sysplex resources. This value can be configured using the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement. Valid values are in the range 10 - 3600 seconds; the default value is 60 seconds.

Recovery

Indicates the action that is to be taken when a sysplex problem is detected.

The value `Yes` indicates that when a problem is detected, the stack issues messages about the problem, leaves the sysplex group, and deactivates all DVIPA resources that are owned by this stack; the VIPADYNAMIC configuration is restored if the stack rejoins the sysplex group. The default value is `No`. The value `Yes` can be configured by specifying the RECOVERY keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

The value `No` indicates that when a problem is detected, the stack issues messages regarding the problem but takes no other action. The value `No` can be configured by specifying the NORECOVERY keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

DelayJoin

Indicates whether the TCP/IP stack delays joining the sysplex group during stack initialization or rejoining the sysplex group following a VARY TCPIP,,OBEYFILE command.

The value `No` indicates that TCP/IP immediately joins the sysplex group during stack initialization. The default value is `No` and can be configured by specifying the NODELAYJOIN keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

The value `Yes` indicates that TCP/IP delays joining the sysplex group during stack initialization until the following conditions true:

- OMPROUTE is started and active
- At least one of monitored interfaces is defined and active (if MONINTERFACE is configured)
- At least one dynamic route over the monitored interfaces is available (if MONINTERFACE DYNROUTE is configured)

Any sysplex-related definitions within the TCP/IP profile (for example, VIPADYNAMIC or IPCONFIG/IPCONFIG6 DYNAMICXCF statements) are not processed until the sysplex group is joined. The value `Yes` can be configured by specifying the DELAYJOIN keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

Join Indicates whether the TCP/IP stack joins the sysplex group during stack initialization.

The value Yes indicates that the TCP/IP stack immediately attempts to join the sysplex group during stack initialization. This is the default setting.

The value No indicates that the TCP/IP stack does not join the sysplex group during stack initialization. You can configure the value No by specifying the NOJOIN keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

If NOJOIN is configured, the TCP/IP stack does not process any VIPADYNAMIC block or DYNAMICXCF statements. Any other GLOBALCONFIG SYSPLEXMONITOR parameter settings (configured or default) are ignored, and the settings are saved in case you want the TCP/IP stack to join the sysplex group at a later time.

If you subsequently issue a VARY TCPIP,,SYSPLEX,JOINGROUP command, the NOJOIN setting is overridden and the saved GLOBALCONFIG SYSPLEXMONITOR parameter settings become active. For example, if you configure NOJOIN and DELAYJOIN, DELAYJOIN is initially ignored. After you issue a V TCPIP,,SYSPLEX,JOINGROUP command, NOJOIN is overridden, DELAYJOIN becomes active, and the stack joins the sysplex group if OMPROUTE is initialized.

Any sysplex-related definitions within the TCP/IP profile, such as VIPADYNAMIC or IPCONFIG DYNAMICXCF statements, are not processed until the TCP/IP stack joins the sysplex group.

MonIntf

Indicates whether the TCP/IP stack is monitoring the status of specified network interfaces.

The value No indicates that the TCP/IP stack is not monitoring the status of network interfaces. The default value is No and it can be configured by specifying the NOMONINTERFACE keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

The value Yes indicates that the TCP/IP stack is monitoring the status of network interfaces that have the MONSYSPLEX attribute specified on the LINK or INTERFACE profile statement. The value Yes can be configured by specifying the MONINTERFACE keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

DynRoute

Indicates whether the TCP/IP stack is monitoring the presence of dynamic routes over the monitored network interfaces.

The value No indicates that the TCP/IP stack is not monitoring the presence of dynamic routes over monitored network interfaces. The default value is No and it can be configured by specifying the NODYNROUTE keyword for the SYSPLEXMONITOR MONINTERFACE parameter on the GLOBALCONFIG profile statement.

The value Yes indicates that the TCP/IP stack is monitoring the presence of dynamic routes over monitored network interfaces that have the MONSYSPLEX attribute specified on the LINK or

INTERFACE statement. It can be configured by specifying the DYNROUTE keyword for the SYSPLEXMONITOR MONINTERFACE parameter on the GLOBALCONFIG profile statement.

AutoRejoin
Indicates whether the TCP/IP stack automatically rejoins the sysplex group when all detected problems that caused the stack to leave the group are relieved.

The value `No` indicates that the stack does not rejoin the group or restore its VIPADYNAMIC definitions when all detected problems have been relieved. The default value is `No` and it can be configured by specifying the NOAUTOREJOIN keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

The value `Yes` indicates that the stack automatically rejoins the sysplex group and restores all of its VIPADYNAMIC configuration definitions. The value `Yes` can be configured by specifying the AUTOREJOIN keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

Restriction: You can specify the AUTOREJOIN keyword only if the RECOVERY keyword is also specified (or is currently enabled) on the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

zIIP Displays information about displacing CPU cycles for various functions onto a System z Information Integration Processor (zIIP). The value Yes for a function indicates that cycles can be displaced to a zIIP when at least one zIIP device is online. Issue the MVS D M=CPU command to display zIIP status. See displaying system configuration information details in z/OS MVS System Commands for more information about displaying processor status.

IPSecurity
Indicates whether the stack is configured to displace CPU cycles for IPSec workload onto a zIIP. This field can have the following values:

Yes Indicates that IPSec CPU cycles are displaced to a zIIP as long as at least one zIIP device is online.

No Indicates that IPSec CPU cycles are not being displaced to a zIIP.

IQDIOMultiWrite
Indicates whether the stack is configured to displace CPU cycles for HiperSockets multiple write workload onto a zIIP. This field can have the following values:

Yes Indicates that the stack is configured to permit HiperSockets multiple write CPU cycles to be displaced to a zIIP.

No Indicates that the stack is configured to not permit HiperSockets multiple write CPU cycles to be displaced to a zIIP.

SMCR

Indicates whether this stack supports Shared Memory Communications over Remote Direct Memory Access (SMC-R) for external data network communications. This field can have the following values:

Yes Indicates that this stack can communicate with other stacks on the external data network by using SMC-R. The SMCR parameter was specified on the GLOBALCONFIG profile statement. When the `SMCR` field has the value `Yes`, the following information is displayed:

FixedMemory

Indicates the maximum amount, in megabytes, of 64-bit private storage that the stack can use for the send and receive buffers that are required for SMC-R communications. The fixed memory value was defined by using the SMCR FIXEDMEMORY parameter on the GLOBALCONFIG. If the SMCR FIXEDMEMORY parameter was not specified in the GLOBALCONFIG statement, the default value of 256 is displayed.

TcpKeepMinInt

Indicates the minimum supported TCP keepalive interval for SMC-R links. Use the SMCR TCPKEEPMININTERVAL parameter on the GLOBALCONFIG statement to define the interval. For applications that are using the TCP_KEEPALIVE setsockopt() option, this interval indicates the minimum interval that TCP keepalive packets are sent on the TCP path of an SMC-R link. The range is 0 - 2147460 seconds. If the interval value is set to 0, TCP keepalive probe packets on the TCP path of an SMC-R link are disabled. If the SMCR TCPKEEPMININTERVAL parameter was not specified in the GLOBALCONFIG statement, then the default interval value of 300 is displayed.

PFID Indicates the Peripheral Component Interconnect Express (PCIe) function ID (PFID) value that was defined using SMCR PFID parameter. The combination of PFID and port number uniquely identifies an 10GbE RoCE Express interface. The stack uses 10GbE RoCE Express features for SMC-R communications with other stacks on the external data network. The PFID is a 2-byte hexadecimal value.

PortNum

Indicates the 10GbE RoCE Express port number that is used for the associated PFID. The PortNum value was specified with the PFID value on the SMCR parameter of the GLOBALCONFIG statement in the TCP/IP profile. The port number can be 1 or 2; the default port is 1.

MTU Indicates the configured maximum transmission unit (MTU) value that is used for the associated PFID. The MTU value can be 1024 or 2048 and the default MTU value is 1024.

No Indicates that this stack cannot communicate with other stacks on the external data network by using SMC-R communications.

The NOSMCR parameter was specified on the GLOBALCONFIG profile statement or the value was set by default.

- **Network Monitor Configuration information**

  Display the following configured network monitor information defined in the NETMONITOR profile statement. For more information about each network monitor parameter, see the NETMONITOR profile statement information in the z/OS Communications Server: IP Configuration Reference.

  PktTrcSrv
  Indicates whether the packet trace service is enabled or disabled. The value `Yes` indicates that PKTTRCSERVICE is in effect (the PKTTRCSERVICE parameter was specified in the NETMONITOR profile statement). The value `No` indicates that PKTTRCSERVICE is not in effect (the NOPKTTRCSERVICE parameter was specified in the NETMONITOR profile statement or is in effect by default).

  TcpCnnSrv
  Indicates whether the TCP connection information service is enabled or disabled. The value `Yes` indicates that TCPCONNSERVICE is in effect (the TCPCONNSERVICE parameter was specified in the NETMONITOR profile statement). The value `No` indicates that TCPCONNSERVICE is not in effect (the NOTCPCONNSERVICE parameter was specified in the NETMONITOR profile statement or is in effect by default).

  MinLifTim
  The minimum lifetime for a new TCP connection to be reported by the service when the TCP connection information service is enabled. If the NOTCPCONNSERVICE parameter was specified in the NETMONITOR profile statement or is in effect by default, then the MinLifTim field is not displayed.

  NtaSrv
  Indicates whether the OSAENTA trace service is enabled or disabled. The value `Yes` indicates that NTATRCSERVICE is in effect (the NTATRCSERVICE parameter was specified in the NETMONITOR profile statement). The value `No` indicates that NTATRCSERVICE is not in effect (the NONTATRCSERVICE parameter was specified in the NETMONITOR profile statement or is in effect by default).

  SmfSrv
  Indicates whether the real-time SMF information service is enabled or disabled. The value `Yes` indicates that SMFSERVICE is enabled (the SMFSERVICE parameter was specified in the NETMONITOR profile statement). The value `No` indicates that SMFSERVICE is disabled (the NOSMFSERVICE parameter was specified in the NETMONITOR profile statement or is disabled by default).

  IPSecurity
  Indicates whether the real-time SMF service is providing IPSec SMF records. The value `Yes` indicates that IPSec SMF records are being provided (either the SMFSERVICE parameter was specified with the IPSECURITY subparameter on the NETMONITOR profile statement or the SMFSERVICE parameter was specified without any subparameters). The value `No` indicates that IPSec SMF records are not being provided (the SMFSERVICE parameter was specified with the NOIPSECURITY subparameter on the NETMONITOR profile statement). This field is displayed only if the SmfSrv value is `Yes`.

Profile

Indicates whether the real-time SMF service is providing TCP/IP profile SMF records. The value `Yes` indicates that TCP/IP profile SMF records are being provided (either the SMFSERVICE parameter was specified with the PROFILE subparameter on the NETMONITOR profile statement, or the SMFSERVICE parameter was specified without any subparameters). The value `No` indicates that TCP/IP profile SMF records are not being provided (the SMFSERVICE parameter was specified with the NOPROFILE subparameter on the NETMONITOR profile statement). This field is displayed only if the SmfSrv value is `Yes`.

CSSMTP

Indicates whether the real-time SMF service is providing CSSMTP SMF 119 records for subtype 48, 49, 51 and 52. The value YES indicates that CSSMTP SMF records are being provided (either the SMFSERVICE parameter was specified with the CSSMTP subparameter on the NETMOINTOR profile statement or the SMFSERVICE parameter was specified without any sub parameters). The value NO indicates that CSSMTP SMF records are not being provided (the SMFSERVICE parameter was specified with the NOCSSMTP subparameter on the NETMONITOR profile statement). This field is displayed only if the SMFSrv value is YES.

CSMAIL

Indicates whether the real-time SMF service is providing CSSMTP SMF 119 records for subtype 50. The value YES indicates that CSSMTP SMF mail records are being provided (either the SMFSERVICE parameter was specified with the CSSMTP subparameter on the NETMOINTOR profile statement or the SMFSERVICE parameter was specified without any subparameters). The value NO indicates that CSSMTP SMF mail records are not being provided (the SMFSERVICE parameter was specified with the NOCSSMTP subparameter on the NETMONITOR profile statement). This field is displayed only if the SMFSrv value is YES.

DVIPA

Indicates whether the real-time SMF service is providing sysplex event SMF records. The value `Yes` indicates that sysplex event SMF records are being provided (either the SMFSERVICE parameter was specified with the DVIPA subparameter on the NETMONITOR profile statement, or the SMFSERVICE parameter was specified without any subparameters). The value `No` indicates that sysplex event SMF records are not being provided (the SMFSERVICE parameter was specified with the NODVIPA subparameter on the NETMONITOR profile statement). This field is displayed only if the SmfSrv value is `Yes`.

- **Autolog Configuration Information**

WaitTime

The time, displayed in seconds, that is specified on the AUTOLOG statement that represents the length of time TCP/IP waits for a procedure to stop if the procedure is still active at startup and TCP/IP is attempting to start the procedure again. The procedure could still be active if it did not stop when TCP/IP was last shut down.

ProcName
The procedure that the TCP/IP address space starts.

JobName
The job name used for the PORT reservation statement. The job name might be identical to the procedure name; however, for z/OS UNIX jobs that spawn listener threads, the names are not the same.

ParmString
A string to be added following the START ProcName value. The ParmString value can be up to 115 characters in length and can span multiple lines. If the PARMSTRING parameter on the AUTOLOG profile statement was not specified or if the *parm_string* value was specified with a blank string, then this field displays blanks.

DelayStart
Indicates whether TCP/IP delays starting this procedure until the TCP/IP stack has completed one or more processing steps. This field can have the following values:

Yes Indicates that the TCP/IP stack does not start this procedure until it has completed all of the processing steps identified by the following subparameters:

DVIPA
TCP/IP delays starting this procedure until after the TCP/IP stack has joined the sysplex group and processed its dynamic VIPA configuration (DELAYSTART was specified on the entry for this procedure in the AUTOLOG profile statement with no additional subparameters, or DELAYSTART was specified with the DVIPA subparameter).

TTLS TCP/IP delays starting this procedure until after the Policy Agent has successfully installed the AT-TLS policy in the TCP/IP stack and AT-TLS services are available (DELAYSTART was specified with the TTLS subparameter on the entry for this procedure in the AUTOLOG profile statement).

No Indicates that this procedure is started when TCP/IP is started (DELAYSTART was not specified on the entry for this procedure in the AUTOLOG profile statement).

- **Data Trace Settings if socket data trace is on**

JobName
The application address space name specified on the DATTRACE command or asterisk (*), if not specified.

TrRecCnt
The number of packets traced for this DATTRACE command.

Length
The value of the ABBREV keyword of the DATTRACE command or FULL to capture the entire packet.

IpAddr
The IP address from the IP keyword of the DATTRACE command or asterisk (*), if not specified.

SubNet
The subnet mask from the SUBNET keyword of the DATTRACE command or asterisk (*), if not specified.

PrefixLen
The prefix length specified on the DATTRACE command.

PortNum
The port number from the PORTNUM keyword of the DATTRACE command or an asterisk (*), if a value was not specified.

Netstat COnn/-c report

Displays the information about each active TCP connection and UDP socket. COnn/**-c** is the default parameter.

TSO syntax:

```
►►──NETSTAT COnn───┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────►◄
```

Modifier:

```
     ┌────────────────┐
     ▼                │
►►──┬─APPLDATA─┬──────┴──────────────────────────────────────────────────────►◄
    └─SERVER───┘
```

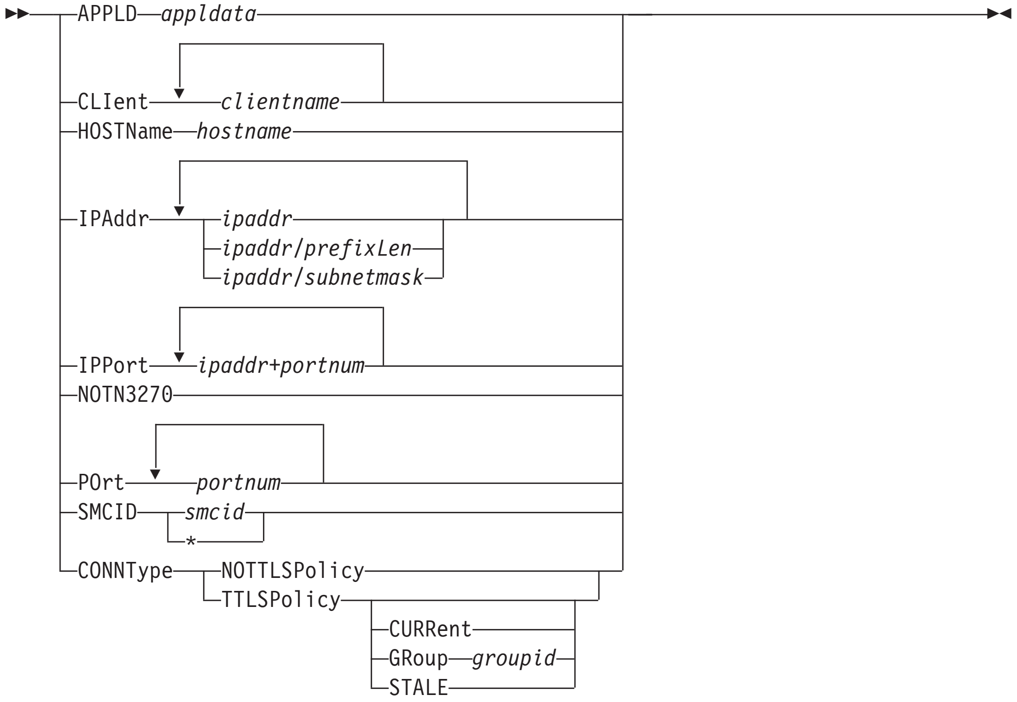
APPLDATA
Provides application data in the output report.

SERVER
Provide detailed information only for TCP connections in the listen state.

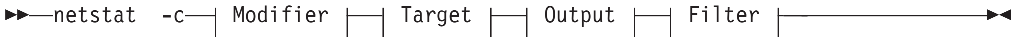
Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

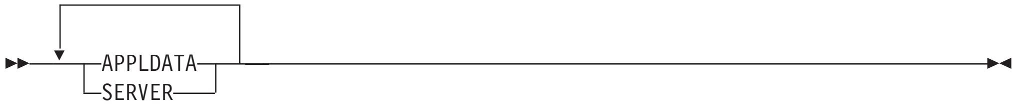
Filter:



z/OS UNIX syntax:



Modifier:



APPLDATA
Provides application data in the output report.

SERVER
Provide detailed information only for TCP connections in the listen state.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
             .---------------.
             V               |
>>-+- -B--------ipaddr+portnum-+-----+----------------------------><
   |         .-------------.         |
   |         V             |         |
   +- -E--------clientname-+---------+
   +- -G--appldata-------------------+
   +- -H--hostname-------------------+
   |         .---------------------. |
   |         V                     | |
   +- -I-------+-ipaddr------------+-+-+
   |           +-ipaddr/prefixLen--+   |
   |           '-ipaddr/subnetmask-'   |
   |         .-----------.             |
   |         V           |             |
   +- -P--------portnum--+-------------+
   +- -T-------------------------------+
   +- -U--+-smcid-+--------------------+
   |      '-*-----'                    |
   '- -X--+-NOTTLSPolicy------------------+-'
          '-TTLSPolicy--+---------------+-'
                        +-CURRent-------+
                        +-GRoup--groupid+
                        '-STALE---------'
```

Filter description:

APPLD/-G *appldata*
: Filter the output of the COnn/**-c** report using the specified application data *appldata*. You can enter one filter value at a time; the specified value can be up to 40 characters in length.

CLIent/-E *clientname*
: Filter the output of the COnn/**-c** report using the specified client name *clientname*. You can enter up to six filter values; each specified value can be up to eight characters in length.

HOSTName/-H *hostname*
: Filter the output of the COnn/**-c** report using the specified host name *hostname*. You can enter one filter value at a time; the specified value can be up to 255 characters in length.

Result: At the end of the report, Netstat displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver which it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value depending on the resolver and DNS configuration.

IPAddr/-I *ipaddripaddr/prefixlengthipaddr/subnetmask*
: Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values; each specified IPv4 *ipaddr* value can be up to 15 characters in length.

ipaddr Filter the output of the COnn/**-c** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* value of 128 is used.

ipaddr/prefixlength
Filter the output of the COnn/**-c** report using the specified IP address and prefix length *ipaddr/prefixlength*. For a IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask
Filter the output of the COnn/**-c** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be either the local or remote IP address.
2. For an IPv6 enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as its IPv4 address does.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

IPPort/-B *ipaddr***+***portnum*
Filter the report output of the COnn/**-c** report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be either the local or remote IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.

- For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.

NOTN3270/-T
Filter the output of the COnn/**-c** report excluding TN3270 server connections.

POrt/-P *portnum*
Filter the output of the COnn/**-c** report using the specified port number *portnum*. You can enter up to six filter values.

Guideline: The port number can be either a local or remote port.

Restriction: For a UDP endpoint socket, the filter value applies only to the local or source port.

SMCID/-U *smcid*
Filter the output of the COnn/-c report by using the specified Shared Memory Communications over Remote Direct Memory Access (SMC-R) link or link group identifier *smcid*. If an asterisk (*) is specified for the filter value, Netstat provides output only for entries that are associated with SMC-R links, and link groups. You can enter one filter value at a time.

CONNType/-X
Filter the report using the specified connection type. You can enter one filter value at a time.

NOTTLSPolicy
Filter the output of the COnn/**-c** report, displaying only connections that have not been matched to an Application Transparent Transport Layer Security (AT-TLS) rule. This includes connections that were established while the AT-TLS function was disabled (NOTTLS was specified on the TCPCONFIG statement or in effect by default) and all connections that do not use the TCP protocol. For TCP connections that were established while the AT-TLS function was enabled, this includes:

- Connections for which AT-TLS policy lookup has not yet occurred (typically the first send or receive has not been issued yet)
- Connections for which AT-TLS policy lookup has occurred but no matching rule was found

TTLSPolicy
Filter the output of the COnn/**-c** report, displaying only connections that match an Application Transparent Transport Layer Security (AT-TLS) rule. This includes only connections that were established while the AT-TLS function was enabled, for which an AT-TLS policy rule was found with either `TTLSEnabled ON` or `TTLSEnabled OFF` specified in the TTLSGroupAction. Responses can be further limited on AT-TLS connection type. AT-TLS connection type has the following values:

CURRent
Display only connections that are using AT-TLS where the rule and all actions are still available to be used for new connections.

GRoup *groupid*
Display only connections that are using the AT-TLS group

that is specified by the *groupid* value. The specified *groupid* value is a number that is assigned by the TCP/IP stack that uniquely identifies an AT-TLS group. You can determine the *groupid* value from the GroupID field value that is displayed in the Netstat TTLS/**-x** GROUP report.

STALE

Display only connections that are using AT-TLS where the rule or at least one action is no longer available to be used for new connections.

The filter value for CLIent/**-E**, IPAddr/**-I**, and APPLD/-G can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/**-I** filter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/prefixlen* or *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -c -I '10.*.0.0'** or **netstat -c -I "10.*.0.0"**.

Command syntax examples:

From TSO environment:

```
NETSTAT CONN
   Display information for all active TCP connections and UDP sockets in the default TCP/IP
   stack.
NETSTAT CONN TCP TCPCS6
   Display information for all active TCP connections and UDP sockets in TCPCS6 stack.
NETSTAT CONN TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display information for these active TCP connections and UDP sockets in TCPCS8 stack
   whose local or remote IP addresses match the specified filter IP address values.
NETSTAT CONN (PORT 2222 6666 88
   Display information for those active TCP connections and UDP sockets in the default
   TCP/IP stack whose local or remote ports match the specified filter port numbers.
```

From UNIX shell environment:

```
   netstat -c
   netstat -c -p tcpcs6
   netstat -c -p tcpcs6 -I 9.43.1.1 9.43.2.2
   netstat -c -P 2222 6666 88
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT CONN
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
User Id  Conn     Local Socket           Foreign Socket         State
-------  ----     ------------           --------------         -----
FTPD1    0000003B 0.0.0.0..21            0.0.0.0..0             Listen
FTPD1    0000003D 9.37.65.146..21        9.67.115.5..1026       Establsh
FTPD1    0000003F 9.37.65.146..21        9.27.13.21..3711       Establsh
TCPCS    0000000F 0.0.0.0..23            0.0.0.0..0             Listen
TCPCS    0000000C 9.67.115.5..23         9.27.11.182..4886      Establsh
APPV4    00000015 0.0.0.0..2049          9.42.103.99..1234      UDP
SYSLOGD1 00000010 0.0.0.0..514           *..*                   UDP
```

IPv6 enabled or request for LONG format:

```
NETSTAT CONN
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
User Id  Conn     State
-------  ----     -----
FTPD1    0000004A Listen
  Local Socket:   ::..21
  Foreign Socket: ::..0
FTPD1    00000052 Establsh
  Local Socket:   ::ffff:9.67.115.5..21
  Foreign Socket: ::ffff:9.67.115.65..1026
FTPD1    00000058 Establsh
  Local Socket:   2001:0db8::9:67:115:66..21
  Foreign Socket: 2001:0db8::9:67:115:65..1027
TCPCS    0000001A Listen
  Local Socket:   0.0.0.0..23
  Foreign Socket: 0.0.0.0..0
TCPCS    0000001E Establsh
  Local Socket:   9.67.115.5..23
  Foreign Socket: 9.27.11.182..4665
USER3    0000005F Establsh
  Local Socket:   2001:0db8::9:67:115:5..1079
  Foreign Socket: 2001:0db8::9:67:115:65..21
USER6    000000C7 Establsh
  Local Socket:   9.67.115.5..1027
  Foreign Socket: 9.37.65.146..21
APPM     00000017 UDP
  Local Socket:   ::ffff.0.0.0.0..2051
  Foreign Socket: ::ffff.9.42.103.99..1234
APPV4    00000015 UDP
  Local Socket:   0.0.0.0..2049
  Foreign Socket: 9.42.103.99..1234

SYSLOGD1 0000002C UDP
  Local Socket:   0.0.0.0..529
  Foreign Socket: *..*
```

Report field descriptions:

User Id
: See the Client name or User ID information in “Netstat report general concept” on page 324 for a detailed description.

Conn See the Client ID or Connection Number information in “Netstat report general concept” on page 324 for a detailed description.

Local Socket
: See the Local Socket information in “Netstat report general concept” on page 324 for a detailed description.

Foreign Socket
: See the Foreign Socket information in “Netstat report general concept” on page 324 for a detailed description.

State See the TCP connection status and UDP socket status information in "Netstat report general concept" on page 324 for a detailed description.

Application Data
The application data that makes it easy for you to locate and display the connections that are used by the application. The beginning of the application data identifies the format of the application data area. For z/OS Communications Server applications, see application data in the z/OS Communications Server: IP Programmer's Guide and Reference for a description of the format, content, and meaning of the data that is supplied by the application. For other applications, see the documentation that is supplied by the application. The data is displayed in character format if application data is present. Non-printable characters, if any, are displayed as dots.

Netstat DEFADDRT/-l report

Displays the policy table for IPv6 default address selection. See the information about the policy table for IPv6 default address selection in z/OS Communications Server: IPv6 Network and Application Design Guide.

TSO syntax:

```
►►──NETSTAT DEFADDRT──┬────────┬──┬────────┬─────────────────────────────►◄
                      └─Target─┘  └─Output─┘
```

Target:
Provide the report for a specific TCP/IP address space by using the TCp *tcpname* parameter. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user terminal. For other options, see "Netstat command output" on page 316.

z/OS UNIX syntax:

```
►►──netstat -l──┬────────┬──┬────────┬───────────────────────────────────►◄
                └─Target─┘  └─Output─┘
```

Target:
Provide the report for a specific TCP/IP address space by using the **-p** *tcpname* parameter. See "The Netstat command target" on page 316 for more information about the **-p** parameter.

Output:
The default output option displays the output on the user terminal. For other options, see "Netstat command output" on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT DEFADDRT
```

From UNIX shell environment:

```
netstat -l
```

Report examples:
You can generate reports by using the TSO NETSTAT command. The z/OS UNIX command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT DEFADDRT

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:30:49

Policy Table for IPv6 Default Address Selection:
  Source: Configured
Precedence Label Prefix
---------- ----- ------
50         0     ::1/128
40         1     ::/0
30         2     2002::/16
20         3     ::/96
10         4     ::ffff:0.0.0.0/96
```

Report field descriptions:

Source
The source of the data in the report. The data source can be one of the following values:

Configured
The policy table displayed in the report was configured using the DEFADDRTABLE TCP/IP profile statement.

Default
The policy table displayed in the report is the default policy table defined by the TCP/IP stack.

Precedence
An integer value in the range 0 - 65530 that indicates the precedence that is used to sort destination addresses.

Label An integer value in the range 0 - 65530 that indicates that a particular source address prefix is preferred for use with a destination address prefix.

Prefix The address prefix that is used to select the policy table entry that best matches a source address or a destination address. The digits (in colon-hexadecimal format) before the slash (/) indicate the prefix. The digits that follow the slash (/) indicate the length of the prefix, in bits. The prefix length is an integer value in the range 1-128.

Netstat DEvlinks/-d report

Displays information about interfaces that are defined to the TCP/IP stack.

TSO syntax:

```
►►──NETSTAT DEvlinks──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────►◄
```

Modifier:

```
►►──SMC────────────────────────────────────────────────────────────────────►◄
```

SMC Provides only detailed Shared Memory Communications over Remote Direct Memory Access (SMC-R) information about all 10GbE RoCE Express interfaces and their associated SMC-R link groups and SMC-R links.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:

```
►►──┬─INTFName──intfname─┬──────────────────────────────────────────────►◄
    └─SMCID──┬─smcid─┬───┘
             └─*─────┘
```

z/OS UNIX syntax:

```
►►──netstat  -d──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────────►◄
```

Modifier:

```
►►──SMC──────────────────────────────────────────────────────────────────►◄
```

SMC Provide only detailed SMC-R information about all 10GbE RoCE Express interfaces and their associated SMC-R link groups and SMC-R links.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Filter:

```
►►──┬─ -K──intfname───┬─────────────────────────────────────────────────►◄
    └─ -U──┬─smcid─┬──┘
           └─*─────┘
```

Filter description:

INTFName/-K *intfname*
Filter the output of the DEvlinks/**-d** report using the specified interface name *intfname*. You can enter one filter value at a time and the specified value can be up to 16 characters in length.

The INTFName/**-K** filter value *intfname* can be one of the following values:

- The link name of a network interface that was configured on a LINK profile statement (this option selects one interface).
- The interface name of a network interface that was configured on an INTERFACE profile statement (this option selects one interface).
- The interface name of an OSAENTA trace interface, which is EZANTA*portname*, where the *portname* value is the name that is specified on the PORTNAME keyword in the TRLE for the OSA-Express port that is being traced (this option selects one interface).
- The port name of an OSA-Express feature in QDIO mode, where the port name is the name that is specified on the PORTNAME keyword in the TRLE (this option selects all interfaces that are associated with the OSA-Express port, including an OSAENTA trace interface).
- The name of a HiperSockets TRLE. This option selects all interfaces that are associated with the HiperSockets TRLE.

Restriction:

- The INTFName/**-K** filter value does not support wildcard characters.
- The INTFName/**-K** filter value does not display information for a device that does not have a link defined.

SMCID/-U *smcid*

Filter the output of the DEvlinks/-d report by using the specified SMC-R link or link group identifier *smcid*. You can enter one filter value at a time.

- If the filter value is an SMC-R link ID, then the report shows details about that SMC-R link and information about the SMC-R link group to which that link belongs.
- If the filter value is an SMC-R link group ID, then the report shows details about all SMC-R links in the link group and information about the SMC-R link group.
- If the filter value is an asterisk (*), then the report provides the same information that the SMC modifier provides.

Rule: If you specify the SMCID/-U filter on the command, the report is generated as if the SMC modifier was also specified. The report includes detailed information about the SMC-R link that *smcid* defines, regardless of whether the SMC modifier was explicitly coded.

Command syntax examples:

From TSO environment:

```
NETSTAT DEVLINKS
   Displays the information about devices and defined interfaces or links in the default
   TCP/IP address space
NETSTAT DEVLINKS TCP TCPCS6
   Displays the information about devices and defined interfaces or links in the TCPCS6
   TCP/IP address space.
NETSTAT DEVLINKS SMC
   Displays additional SMC-R information about RNIC interfaces.
NETSTAT DEVLINKS TCP TCPCS8 (INTFNAME OSAQDIOLINK
   Display the information for the OSAQDIOLINK in the TCPCS8 TCP/IP adress space.
```

From UNIX shell environment:

```
netstat -d
netstat -d -p tcpcs6
netstat -d SMC
netstat -d -p tcpcs8 -K OSAQDIOLINK
```

Report examples:

The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT DEVLINKS
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:23:39
DevName: LOOPBACK          DevType: LOOPBACK
  DevStatus: Ready
  LnkName: LOOPBACK          LnkType: LOOPBACK   LnkStatus: Ready
    ActMtu: 65535
  Routing Parameters:
    MTU Size: n/a               Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 0.0.0.0
  Multicast Specific:
    Multicast Capability: No
  Link Statistics:
    BytesIn                           = 24943
    Inbound Packets                   = 100
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 24943
    Outbound Packets                  = 100
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

DevName: LCS1              DevType: LCS       DevNum: 0D00
  DevStatus: Ready
  LnkName: TR1               LnkType: TR          LnkStatus: Ready
    NetNum: 0    QueSize: 0
    MacAddrOrder: Non-Canonical    SrBridgingCapability: Yes
    IpBroadcastCapability: Yes     ArpBroadcastType: All Rings
    MacAddress: 08005A0D97A2
    ActMtu: 1492
    SecClass: 8                      MonSysplex: Yes
  Routing Parameters:
    MTU Size: 02000             Metric: 100
    DestAddr: 0.0.0.0           SubnetMask: 255.255.255.128
  Packet Trace Setting:
    Protocol: *                 TrRecCnt: 00000006  PckLength: FULL
    Discard : NONE
    SrcPort: *                  DestPort: *         PortNum: *
    IpAddr: *                   SubNet: *
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt      SrcFltMd
    -----             ------      --------
    224.0.0.1         0000000001  Include
      SrcAddr: 9.1.1.1
               9.1.1.2
               9.1.1.3
    224.9.9.3         0000000001  Include
      SrcAddr: 9.1.1.1
    224.9.9.4         0000000001  Exclude
      SrcAddr: 9.2.2.1
               9.2.2.2
    225.9.9.4         0000000003  Exclude
      SrcAddr: None
  Link Statistics:
    BytesIn                           = 9130
    Inbound Packets                   = 2
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 60392
    Outbound Packets                  = 11
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

DevName: OSAQDIO4          DevType: MPCIPA
  DevStatus: Ready         CfgRouter: Non  ActRouter: Non
  LnkName: OSAQDIOLINK       LnkType: IPAQENET   LnkStatus: Ready
    Speed: 0000000100
    IpBroadcastCapability: No
    VMACAddr:   000629DC21BC  VMACOrigin: Cfg  VMACRouter: All
    ArpOffload: Yes                  ArpOffloadInfo: Yes
    ActMtu: 1492
    VLANid: 1260                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    ReadStorage: GLOBAL (8064K)      InbPerf: Balanced
    ReadStorage: GLOBAL (8064K)
    InbPerf: Balanced
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 8                      MonSysplex: Yes
  Routing Parameters:
    MTU Size: n/a              Metric: 00
    DestAddr: 0.0.0.0          SubnetMask: 255.255.255.192
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt        SrcFltMd
    -----             ------        --------
    224.0.0.1         0000000001    Exclude
      SrcAddr: None
  Link Statistics:
    BytesIn                           = 11476
    Inbound Packets                   = 10
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 6707
    Outbound Packets                  = 10
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
```

```
DevName: OSATRL90          DevType: ATM
  DevStatus: Not Active
  LnkName: OSA90LINK1        LnkType: ATM        LnkStatus: Not Active
    ActMtu: Unknown
    SecClass: 8                      MonSysplex: Yes
  Routing Parameters:
    MTU Size: n/a               Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 255.0.0.0
  ATM Specific:
    ATM portName:  OSA90
    ATM PVC Name:  STEPH             PVC Status: Not Active

    ATM LIS Name:  LIS1
    SubnetValue:   9.67.1.0          SubnetMask:    255.255.255.0
    DefaultMTU:    0000009180        InactvTimeOut: 0000000300
    MinHoldTime:   0000000060        MaxCalls:      0000001000
    CachEntryAge:  0000000900        ATMArpReTry:   0000000002
    ATMArpTimeOut: 0000000003        PeakCellRate:  0000000000
    NumOfSVCs:     0000000000        BearerClass:   C

    ATMARPSV Name: ARPSV1
    VcType:        PVC               ATMaddrType: NSAP
    ATMaddr:
    IpAddr:        0.0.0.0
  Multicast Specific:
    Multicast Capability: No
  Link Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

DevName: CLAW2             DevType: CLAW      DevNum: 0D10
  DevStatus: Ready         CfgPacking: Packed ActPacking: Packed
  LnkName: CLAW2LINK         LnkType: CLAW       LnkStatus: Ready
    ActMtu: 2600
    SecClass: 8                      MonSysplex: No
  Routing Parameters:
    MTU Size: n/a               Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 255.255.255.0
  Multicast Specific:
    Multicast Capability: No
  Link Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
```

```
DevName: IUTIQDIO          DevType: MPCIPA
  DevStatus: Ready
  LnkName: IQDIOLNK0A3D0001  LnkType: IPAQIDIO   LnkStatus: Ready
    IpBroadcastCapability: No
    CfgRouter: Non                   ActRouter: Non
    ArpOffload: Yes                  ArpOffloadInfo: No
    ActMtu: 8192
    ReadStorage: GLOBAL (2048K)
    SecClass: 255
    IQDMultiWrite: Enabled
  Routing Parameters:
    MTU Size: 8192              Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 255.255.0.0
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt        SrcFltMd
    -----             ------        --------
    224.0.0.1         0000000001    Exclude
      SrcAddr: None
  Link Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

IntfName: OSAQDIOINTF       IntfType: IPAQENET   IntfStatus: Ready
    PortName: OSAQDIO2  Datapath: 0E2A     DatapathStatus: Ready
    CHPIDType: OSD   SMCR: Yes
    PNetID: NETWORK3
    Speed: 0000000100
    IpBroadcastCapability: No
    VMACAddr:   020629DC21BD  VMACOrigin: Cfg  VMACRouter: All
    SrcVipaIntf: VIPAV4
    CfgRouter: Non                   ActRouter: Non
    ArpOffload: Yes                  ArpOffloadInfo: Yes
    CfgMtu: 1492                     ActMtu: 1492
    IpAddr: 100.1.1.1/24
    VLANid: 1261                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    ReadStorage: GLOBAL (8064K)      InbPerf: Balanced
    ReadStorage: GLOBAL (8064K)
    InbPerf: Dynamic
      WorkloadQueueing: Yes
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 9                      MonSysplex: Yes
    Isolate: Yes                     OptLatencyMode: Yes
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt        SrcFltMd
    -----             ------        --------
    224.0.0.1         0000000001    Exclude
  SrcAddr: None
  Interface Statistics:
    BytesIn                           = 12834
    Inbound Packets                   = 16
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 5132
    Outbound Packets                  = 10
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
  Associated RNIC interface: EZARIUT10005
  Associated RNIC interface: EZARIUT10006
```

```
IntfName: IQDINTF1          IntfType: IPAQIDIO   IntfStatus: Ready
    TRLE: IUTIQ4QD  Datapath: 0E2A     DatapathStatus: Ready
    CHPID: D1
    IpBroadcastCapability: No
    SrcVipaIntf: VIPAV4
    ArpOffload: Yes                  ArpOffloadInfo: No
    CfgMtu: 8192                     ActMtu: 8192
    IpAddr: 100.1.1.1/24
    VLANid: 1261
    ReadStorage: GLOBAL (2048K)
    SecClass: 255
    IQDMultiWrite: Enabled
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt        SrcFltMd
    -----             ------        --------
    224.0.0.1         0000000001    Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
  IntfName: VDEV1             IntfType: VIPA       IntfStatus: Ready
    IpAddr: 100.1.1.1/24
  Multicast Specific:
    Multicast Capability: No
```

```
IntfName: OSXC9INT1         IntfType: IPAQENET   IntfStatus: Ready
    PortName: IUTXP0C9  Datapath: 0E56     DatapathStatus: Ready
    CHPIDType: OSX      CHPID: C9
    PNetID: IEDN
    Speed: 0000001000
    IpBroadcastCapability: No
    VMACAddr: 420001AA0E56  VMACOrigin: OSA  VMACRouter: All
    CfgRouter: Non                 ActRouter: Non
    ArpOffload: Yes                ArpOffloadInfo: No
    CfgMtu: None                   ActMtu: 8992
    IpAddr: 172.16.0.1/16
    VLANid: 401                    VLANpriority: Disabled
    DynVLANRegCfg: No              DynVLANRegCap: Yes
    ReadStorage: GLOBAL (512K)
    InbPerf: Dynamic
      WorkloadQueueing: No
    ChecksumOffload: No            SegmentationOffload: No
    SecClass: 255                  MonSysplex: No
    Isolate: No                    OptLatencyMode: No
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt      SrcFltMd
    -----             ------      --------
    224.0.0.1         0000000001  Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
  Associated IQDX interface: EZAIQXC9  IQDX Status: Ready
    BytesIn                           = 0
    Inbound Packets                   = 0
    BytesOut                          = 0
    Outbound Packets                  = 0


IntfName: EZAIQXC9          IntfType: IPAQIQDX    IntfStatus: Ready
    Datapath: 0E0E     DatapathStatus: Ready
    VMACAddr: 820001AA0E0E
    ReadStorage: MAX (2048K)
    IQDMultiWrite: Disabled
  Multicast Specific:
    Multicast Capability: No
  Interface Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
```

```
IntfName: EZARIUT10005      IntfType: RNIC       IntfStatus: Ready
  PFID: 0005  PortNum: 1  TRLE: IUT10005
  PNetID: NETWORK3
  VMACAddr: 02000012F030
  GIDAddr:  fe80::200:ff:fe12:f030
  Interface Statistics:
    BytesIn                          = 18994
    Inbound Operations               = 146
    BytesOut                         = 19139
    Outbound Operations              = 811
    SMC Links                        = 2
    TCP Connections                  = 1
    Intf Receive Buffer Inuse        = 64K
IntfName: EZARIUT10006      IntfType: RNIC       IntfStatus: Ready
  PFID: 0006  PortNum: 1  TRLE: IUT10006
  PNetID: NETWORK3
  VMACAddr: 02000012EF50
  GIDAddr:  fe80::200:ff:fe12:ef50
  Interface Statistics:
    BytesIn                          = 226
    Inbound Operations               = 4
    BytesOut                         = 29
    Outbound Operations              = 4
    SMC Links                        = 2
    TCP Connections                  = 1
    Intf Receive Buffer Inuse        = 64K

IPv4 LAN Group Summary
 LanGroup: 001

   Name             Status      ArpOwner          VipaOwner
   -------          ------      --------          ---------
   OSXC9INT1        Active      OSXC9INT1         Yes
   TR1              Active      TR1               No

 LanGroup: 002

   Name             Status      ArpOwner          VipaOwner
   -------          ------      --------          ---------
   OSAQDIOLINK      Active      OSAQDIOLINK       Yes
   OSAQDIOINTF      Active      OSAQDIOINTF       No
```

IPv6 enabled or request for LONG format:

```
NETSTAT DEVLINKS
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:23:39
DevName: LOOPBACK          DevType: LOOPBACK
  DevStatus: Ready
  LnkName: LOOPBACK          LnkType: LOOPBACK   LnkStatus: Ready
    ActMtu: 65535
  Routing Parameters:
    MTU Size: n/a               Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 0.0.0.0
  Multicast Specific:
    Multicast Capability: No
  Link Statistics:
    BytesIn                           = 7665
    Inbound Packets                   = 100
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 7665
    Outbound Packets                  = 100
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

IntfName: LOOPBACK6        IntfType: LOOPBACK6 IntfStatus: Ready
    ActMtu: 65535
  Multicast Specific:
    Multicast Capability: No
  Interface Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

DevName: LCS1              DevType: LCS       DevNum: 0D00
  DevStatus: Ready
  LnkName: TR1               LnkType: TR          LnkStatus: Ready
    NetNum: 0    QueSize: 0
    MacAddrOrder: Non-Canonical    SrBridgingCapability: Yes
    IpBroadcastCapability: Yes     ArpBroadcastType: All Rings
    MacAddress: 08005A0D97A2
    ActMtu: 1492
    SecClass: 8                      MonSysplex: Yes
  Routing Parameters:
    MTU Size: 02000             Metric: 100
    DestAddr: 0.0.0.0           SubnetMask: 255.255.255.128
  Packet Trace Setting:
    Protocol: *                 TrRecCnt: 00000006  PckLength: FULL
    Discard : NONE
    SrcPort: *                  DestPort: *         PortNum: *
    IpAddr: *                   SubNet: *
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt      SrcFltMd
    -----             ------      --------
    224.9.9.1         0000000002  Include
      SrcAddr: 9.1.1.1
               9.1.1.2
               9.1.1.3
    224.9.9.3         0000000001  Include
      SrcAddr: 9.1.1.1
    224.9.9.4         0000000001  Exclude
      SrcAddr: 9.2.2.1
               9.2.2.2
    225.9.9.4         0000000003  Exclude
      SrcAddr: None
  Link Statistics:
    BytesIn                           = 9130
    Inbound Packets                   = 2
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 60392
    Outbound Packets                  = 11
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

DevName: OSAQDIO4          DevType: MPCIPA
  DevStatus: Ready
  LnkName: OSAQDIOLINK       LnkType: IPAQENET   LnkStatus: Ready
    Speed: 0000000100
    IpBroadcastCapability: No
    VMACAddr:   000629DC21BC  VMACOrigin: Cfg  VMACRouter: All
    CfgRouter: Non                   ActRouter: Non
    ArpOffload: Yes                  ArpOffloadInfo: Yes
    ActMtu: 1492
    VLANid: 1260                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    ReadStorage: GLOBAL (8064K)      InbPerf: Balanced
    ReadStorage: GLOBAL (8064K)
    InbPerf: Balanced
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 8                      MonSysplex: Yes
  Routing Parameters:
    MTU Size: n/a               Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 255.255.255.192
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt        SrcFltMd
    -----             ------        --------
    224.0.0.1         0000000001    Exclude
      SrcAddr: None
  Link Statistics:
    BytesIn                           = 11476
    Inbound Packets                   = 10
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 6707
    Outbound Packets                  = 10
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
```

```
IntfName: OSAQDIO46        IntfType: IPAQENET6   IntfStatus: Ready
    PortName: OSAQDIO4  Datapath: 0E2B     DatapathStatus: Ready
    CHPIDType: OSD      SMCR: Yes
    PNetID: NETWORK3
    QueSize: 0  Speed: 0000000100
    VMACAddr:   000629DC21BC  VMACOrigin: Cfg  VMACRouter: All
    SrcVipaIntf: VIPAV6
    DupAddrDet: 1
    CfgRouter: Pri                   ActRouter: Pri
    RtrHopLimit: 5
    CfgMtu: 4096                     ActMtu: 1492
    VLANid: 1261                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    IntfID: 0000:0000:0000:0001
    ReadStorage: GLOBAL (8064K)
    InbPerf: Balanced
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 8                      MonSysplex: Yes
    Isolate: Yes                     OptLatencyMode: Yes
    TempPrefix: 2001:0db8:3454:a3cf::/64
                2001:0db8:58cd::/48

  Packet Trace Setting:
    Protocol: *                 TrRecCnt: 00000000  PckLength: FULL
    SrcPort:  *                 DestPort: *
    IpAddr/PrefixLen: 9::44/128
  Multicast Specific:
    Multicast Capability: Yes
    Group:     ff02::1:ff15:5
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: 2e00::11
               2e00::22
    Group:     ff02::1:ffdc:217c
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff00:2
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                           = 12655
    Inbound Packets                   = 12
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 4590
    Outbound Packets                  = 11
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
  Associated RNIC interface: EZARIUT10005
  Associated RNIC interface: EZARIUT10006
```

```
IntfName: V6SAMEH          IntfType: MPCPTP6   IntfStatus: Not Active
    TRLE: IUTSAMEH   DevStatus: Not Active
    SrcVipaIntf: VIPAV6
    ActMtu: Unknown
    IntfID: 0000:0000:0000:0001
    SecClass: 8
  Multicast Specific:
    Multicast Capability: No
  Interface Statistics:
    BytesIn                          = 0
    Inbound Packets                  = 0
    Inbound Packets In Error         = 0
    Inbound Packets Discarded        = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                         = 0
    Outbound Packets                 = 0
    Outbound Packets In Error        = 0

IntfName: VIPAV6           IntfType: VIPA6     IntfStatus: Ready
  Packet Trace Setting:
    Protocol: *                 TrRecCnt: 00000000  PckLength: FULL
    SrcPort:  *                 DestPort: *         PortNum: *
    IpAddr:   *                 SubNet:   *
  Multicast Specific:
    Multicast Capability: No
```

```
IntfName: SZQIDIO6          IntfType: IPAQIDIO6  IntfStatus: Not Active
    TRLE: IUTIQDD1  Datapath: 0E3A     DatapathStatus: Not Active
    CHPID: D1
    ActMtu: Unknown
    VLANid: 3
    SecClass: 044                    MonSysplex: No
  Multicast Specific:
    Multicast Capability: Unknown
    Group:     ff02::1:ff00:2
      RefCnt:  0000000002  SrcFltMd: Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                          = 0
    Inbound Packets                  = 0
    Inbound Packets In Error         = 0
    Inbound Packets Discarded        = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                         = 0
    Outbound Packets                 = 0
    Outbound Packets In Error        = 0
```

```
IntfName: OSAQDIOINTF       IntfType: IPAQENET   IntfStatus: Ready
    PortName: OSAQDIO2  Datapath: 0E2A     DatapathStatus: Ready
    CHPIDType: OSD         SMCR: Yes
    PNetID: ZOSNET
    Speed: 0000000100
    IpBroadcastCapability: No
    VMACAddr:   020629DC21BD  VMACOrigin: Cfg  VMACRouter: All
    SrcVipaIntf: VIPAV4
    CfgRouter: Non                   ActRouter: Non
    ArpOffload: Yes                  ArpOffloadInfo: Yes
    CfgMtu: 1492                     ActMtu: 1492
    IpAddr: 100.1.1.1/24
    VLANid: 1261                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    ReadStorage: GLOBAL (8064K)
    InbPerf: Balanced
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 9                      MonSysplex: Yes
    Isolate: Yes
  Multicast Specific:
    Multicast Capability: Yes
    Group             RefCnt        SrcFltMd
    -----             ------        --------
    224.0.0.1         0000000001    Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                           = 12834
    Inbound Packets                   = 16
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 5132
    Outbound Packets                  = 10
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
  Associated RNIC interface: EZARIUT10005
  Associated RNIC interface: EZARIUT10006
```

```
IntfName: OSXC9INT2         IntfType: IPAQENET6  IntfStatus: Ready
    PortName: IUTXP0C9  Datapath: 0E56     DatapathStatus: Ready
    CHPIDType: OSX      CHPID: C9
    PNetID: IEDN
    QueSize: 0    Speed: 0000001000
    VMACAddr: 620001AA0E56   VMACOrigin: OSA    VMACRouter: All
    DupAddrDet: 1
    CfgMtu: None                     ActMtu: 9000
    VLANid: 602                      VLANpriority: Disabled
    DynVLANRegCfg: No                DynVLANRegCap: Yes
    ReadStorage: GLOBAL (512K)
    InbPerf: Dynamic
      WorkloadQueueing: No
    ChecksumOffload: No              SegmentationOffload: No
    SecClass: 255                    MonSysplex: No
    Isolate: No                      OptLatencyMode: No
    TempPrefix: All
  Multicast Specific:
    Multicast Capability: Yes
    Group:     ff02::1:ffaa:e56
      RefCnt:  0000000001   SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff01::1
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff01:1
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff00:2
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff00:1
      RefCnt:  0000000002  SrcFltMd: Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 688
    Outbound Packets                  = 7
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0
  Associated IQDX interface: EZ6IQXC9  IQDX Status: Ready
    BytesIn                           = 0
    Inbound Packets                   = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
```

```
IntfName: EZ6IQXC9          IntfType: IPAQIQDX6   IntfStatus: Ready
    Datapath: 0E0E          DatapathStatus: Ready
    VMACAddr: 820001AA0E0E
    ReadStorage: MAX (2048K)
    IQDMultiWrite: Disabled
  Multicast Specific:
    Multicast Capability: ND only
    Group:     ff02::1:ffaa:e56
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff01:1
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff00:2
      RefCnt:  0000000001  SrcFltMd: Exclude
      SrcAddr: None
    Group:     ff02::1:ff00:1
      RefCnt:  0000000002  SrcFltMd: Exclude
      SrcAddr: None
  Interface Statistics:
    BytesIn                           = 0
    Inbound Packets                   = 0
    Inbound Packets In Error          = 0
    Inbound Packets Discarded         = 0
    Inbound Packets With No Protocol  = 0
    BytesOut                          = 0
    Outbound Packets                  = 0
    Outbound Packets In Error         = 0
    Outbound Packets Discarded        = 0

IntfName: EZARIUT10005      IntfType: RNIC       IntfStatus: Ready
  PFID: 0005  PortNum: 1  TRLE: IUT10005 PFIDStatus: Ready
  PNetID: NETWORK3
  VMACAddr: 02000012F030
  GIDAddr:  fe80::200:ff:fe12:f030
  Interface Statistics:
    BytesIn                           = 18994
    Inbound Operations                = 146
    BytesOut                          = 19139
    Outbound Operations               = 811
    SMC Links                         = 2
    TCP Connections                   = 1
    Intf Receive Buffer Inuse         = 64K

IntfName: EZARIUT10006      IntfType: RNIC       IntfStatus: Ready
  PFID: 0006  PortNum: 1  TRLE: IUT10006 PFIDStatus: Ready
  PNetID: NETWORK3
  VMACAddr: 02000012EF50
  GIDAddr:  fe80::200:ff:fe12:ef50
  Interface Statistics:
    BytesIn                           = 226
    Inbound Operations                = 4
    BytesOut                          = 29
    Outbound Operations               = 4
    SMC Links                         = 2
    TCP Connections                   = 1
    Intf Receive Buffer Inuse         = 64K
```

```
IPv4 LAN Group Summary
 LanGroup: 001

   Name              Status     ArpOwner          VipaOwner
   -------           ------     --------          ---------
   TR1               Active     TR1               No

 LanGroup: 002

   Name              Status     ArpOwner          VipaOwner
   -------           ------     --------          ---------
   OSAQDIOLINK       Active     OSAQDIOLINK       Yes
   OSAQDIOINTF       Active     OSAQDIOINTF       No

IPv6 LAN Group Summary
 LanGroup: 004

   Name              Status     NDOwner           VipaOwner
   --------          ------     --------          ---------
   OSAQDIO46         Active     OSAQDIO46         Yes
```

```
NETSTAT DEVLINKS SMC
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPIP1          12:04:57
IntfName: EZARIUT10005      IntfType: RNIC       IntfStatus: Ready
  PFID: 0005  PortNum: 1  TRLE: IUT10005
  PNetID: NETWORK3
  VMACAddr: 02000012F030
  GIDAddr:  fe80::200:ff:fe12:f030
  Interface Statistics:
    BytesIn                          = 18994
    Inbound Operations               = 146
    BytesOut                         = 19139
    Outbound Operations              = 811
    SMC Links                        = 2
    TCP Connections                  = 1
    Intf Receive Buffer Inuse        = 64K
  SMC Link Information:
    LocalSMCLinkId: FB710601  RemoteSMCLinkId: 72420601
      SMCLinkGroupId: FB710600  VLANid: 100  MTU: 1024
      LocalGID:  fe80::200:ff:fe12:f030
        LocalMACAddr:  02000012F030  LocalQP:  00004B
      RemoteGID: fe80::200:1ff:fe12:f030
        RemoteMACAddr: 02000112F030  RemoteQP: 00004A
      SMCLinkBytesIn:            498
      SMCLinkInOperations:       12
      SMCLinkBytesOut:           294
      SMCLinkOutOperations:      13
      TCP Connections:           0
      Link Receive Buffer Inuse: 0K
        64K   Buffer Inuse:      0K

  SMC Link Information:
    LocalSMCLinkId: FB710701  RemoteSMCLinkId: 72420701
      SMCLinkGroupId: FB710700  VLANid: 100  MTU: 4096
      LocalGID:  fe80::200:ff:fe12:f030
        LocalMACAddr:  02000012F030  LocalQP:  00004C
      RemoteGID: fe80::200:1ff:fe12:f030
        RemoteMACAddr: 02000112F030  RemoteQP: 00004D
      SMCLinkBytesIn:            293
      SMCLinkInOperations:       8
      SMCLinkBytesOut:           490
      SMCLinkOutOperations:      15
      TCP Connections:           1
      Link Receive Buffer Inuse: 64K
        64K   Buffer Inuse:      64K


IntfName: EZARIUT10006      IntfType: RNIC       IntfStatus: Ready
  PFID: 0006  PortNum: 1  TRLE: IUT10006
  PNetID: NETWORK3
  VMACAddr: 02000012EF50
  GIDAddr:  fe80::200:ff:fe12:ef50
  Interface Statistics:
    BytesIn                          = 226
    Inbound Operations               = 4
    BytesOut                         = 29
    Outbound Operations              = 4
    SMC Links                        = 2
    TCP Connections                  = 1
    Intf Receive Buffer Inuse        = 64K
```

```
SMC Link Information:
    LocalSMCLinkId: FB710602  RemoteSMCLinkId: 72420602
      SMCLinkGroupId: FB710600  VLANid: 100  MTU: 2048
      LocalGID:  fe80::200:ff:fe12:ef50
        LocalMACAddr:  02000012EF50  LocalQP:  00004A
      RemoteGID: fe80::200:1ff:fe12:ef50
        RemoteMACAddr: 02000112EF50  RemoteQP: 00004B
      SMCLinkBytesIn:            226
      SMCLinkInOperations:       5
      SMCLinkBytesOut:           29
      SMCLinkOutOperations:      4
      TCP Connections:           1
      Link Receive Buffer Inuse: 64K
        64K   Buffer Inuse:      64K

  SMC Link Information:
    LocalSMCLinkId: FB710702  RemoteSMCLinkId: 72420702
      SMCLinkGroupId: FB710700  VLANid: 100  MTU: 1024
      LocalGID:  fe80::200:ff:fe12:ef50
        LocalMACAddr:  02000012EF50  LocalQP:  00004D
      RemoteGID: fe80::200:1ff:fe12:ef50
        RemoteMACAddr: 02000112EF50  RemoteQP: 00004C
      SMCLinkBytesIn:            0
      SMCLinkInOperations:       0
      SMCLinkBytesOut:           0
      SMCLinkOutOperations:      0
      TCP Connections:           0
      Link Receive Buffer Inuse: 0K
        64K   Buffer Inuse:      0K
SMC Link Group Information:
  SMCLinkGroupId: FB710600  PNetID: NETWORK3
    Redundancy: Full
    Link Group Receive Buffer Total: 3M
      64K   Buffer Total: 1M

    LocalSMCLinkId  RemoteSMCLinkId
    --------------  ---------------
    FB710601        72420601
    FB710602        72420602

  SMCLinkGroupId: FB710700  PNetID: NETWORK3
    Redundancy: Full
    Link Group Receive Buffer Total: 3M
      64K   Buffer Total: 1M

    LocalSMCLinkId  RemoteSMCLinkId
    --------------  ---------------
    FB710701        72420701
    FB710702        72420702
```

Example output for an OSAENTA interface:

```
OSA-Express Network Traffic Analyzer Information:
  OSA PortName: QDIO4101          OSA DevStatus:     Ready
    OSA IntfName: EZANTAQDIO4101  OSA IntfStatus:    Ready
    OSA Speed:    1000            OSA Authorization: Logical Partition
    OSAENTA Cumulative Trace Statistics:
      DataMegs:   0                     Frames:          8
      DataBytes:  760                   FramesDiscarded: 4
      FramesLost: 0
    OSAENTA Active Trace Statistics:
      DataMegs:   0                     Frames:          8
      DataBytes:  760                   FramesDiscarded: 4
      FramesLost: 0                     TimeActive:      8
    OSAENTA Trace Settings:           Status: On
      DataMegsLimit: 1024               FramesLimit:     2147483647
      Abbrev:        224                TimeLimit:       10080
      Discard:       ALL
    OSAENTA Trace Filters:            Nofilter: ALL
      DeviceID: *
      Mac:      *
      VLANid:   *
      ETHType:  *
      IPAddr:   *
      Protocol: *
      PortNum:  *
```

Report field descriptions:

DevName
The device name that is configured on the DEVICE statement.

DevType
The device type that is configured on the DEVICE statement.

DevNum
The device number that is configured on the DEVICE statement. This field is significant only for device types CTC, CLAW, LCS, and CDLC.

DevStatus
The device status. You can use this field if you are having activation problems with the device or interface. Table 16 describes the possible status values:

Table 16. Possible device status values

Device status	Description
Starting	A START of the device has been issued by the operator, and TCP/IP has sent an Activation request to the Data Link Control (DLC) layer.
Sent SETUP	DLC has acknowledged the TCP/IP Activation request, and TCP/IP has requested DLC to perform the initial I/O sequence with the device.
Enabling	DLC has acknowledged the TCP/IP Activation request, and TCP/IP has requested DLC to allow data connections to be established for the device.
Connecting	DLC has accepted the Initial I/O Sequence request.
Connecting2	The control connection for a CLAW device has been established, and the second connection (on which IP traffic is carried) is being established.
Negotiating	The initial I/O sequence with the device is complete, and TCP/IP is performing additional link-layer initialization.
Ready	The initialization sequence with the device is complete. The device is now ready.

Table 16. Possible device status values (continued)

Device status	Description
Deactivating	DLC has performed the first stage of an orderly device deactivation.
Not active	The device is not active. (The device has never been started, or has been stopped after having been started.)

Configured router status (CfgRouter)
The router attribute (PRIROUTER/SECROUTER/NONROUTER) that is specified on the DEVICE or INTERFACE statement. This field is significant only for MPCIPA devices and for IPAQENET and IPAQENET6 interfaces. This field is not displayed if virtual MAC (VMAC) has been configured.

Actual router status (ActRouter)
The router attribute in effect for the device or interface. It matches the configured router status unless the configured value conflicted with the configured value of another stack that is sharing the adapter. This field is significant only for MPCIPA devices and for IPAQENET and IPAQENET6 interfaces. The router attribute is determined when the device or interface starts. This field is not displayed if virtual MAC (VMAC) has been configured.

Virtual MAC address (VMACAddr)
The virtual local hardware address for this link or interface. This field is significant for the following types of devices:

- An IPAQENET link or interface, or an IPAQENET6 interface, where a virtual MAC address was configured by specifying the VMAC parameter. The value `n/a` is displayed if VMAC was configured but a virtual MAC address was not configured.
- An RNIC interface that is created when an IPAQENET or IPAQENET6 interface specified SMCR. The VMAC address is provided by VTAM, and is not configured on the INTERFACE profile statement. `VMACAddr` is displayed for active RNIC interfaces only.

Virtual MAC origin (VMACOrigin)
Displays whether the virtual MAC address (VMACAddr) was configured on the LINK or INTERFACE statement, or was generated by OSA-Express. This field is significant only for IPAQENET links or interfaces and for IPAQENET6 interfaces for which virtual MAC (VMAC) has been configured. The following list shows the possible values:

Cfg The virtual MAC address is configured on the LINK statement or on the INTERFACE statement.

OSA The virtual MAC address has been generated by OSA-Express.

Virtual MAC router status (VMACRouter)
Displays the virtual MAC router attribute that was specified on the LINK or INTERFACE statement using the ROUTEALL or ROUTELCL keywords. This field is significant only for IPAQENET links or interfaces and for IPAQENET6 interfaces for which virtual MAC (VMAC) has been configured. See OSA Routing information in the z/OS Communications Server: IP Configuration Guide for more information about Virtual MAC router attributes. The following list shows the possible values:

All Corresponds to the ROUTEALL keyword. Indicates that all IP traffic destined to the Virtual MAC is forwarded by the OSA-Express device to the TCP/IP stack

Local Corresponds to the ROUTELCL keyword. Indicates that only traffic destined to the Virtual MAC whose destination IP address is registered with the OSA-Express device by this TCP/IP stack is forwarded by the OSA-Express device.

Configured packing status (CfgPacking)
This field is the packing attribute (Packed/None) specified on the DEVICE statement. This field is significant only for CLAW devices.

Actual packing status (ActPacking)
This field indicates the packing attribute in effect for the device. It will match the configured packing status unless packing was requested and the device does not support packing. This field is significant only for a CLAW device and is determined when the device starts.

LnkName/IntfName
This field is the link name or the interface name of the particular device or interface being displayed. If the device or interface is configured, this field is the link name configured in the LINK statement or the interface name configured in the INTERFACE statement. If the link name or interface name is dynamically generated by the TCP/IP stack, this field is the dynamically generated link name or interface name.

LnkType/IntfType
This field is the link type or the interface type of the particular device or interface being displayed. If the device or interface is configured, this field is the link type configured in the LINK statement or the interface type configured in the INTERFACE statement. If the link type or interface type is dynamically generated by the TCP/IP stack, this field is the dynamically generated link type or interface type. A 10GbE RoCE Express interface has an IntfType value equal to RNIC.

LnkStatus/IntfStatus
This field is the link or interface status. The following list describes the possible link or interface status values:

Link/Interface status	Description
Ready	A START of the device/interface has been issued by the operator, and TCP/IP has been sent an Activation request to the Data Link Control (DLC) layer.
Not Active	The link or interface is not active. There is no command to start a link; link activation is normally performed during START device processing. Interface activation is performed during START interface processing. A link or interface is marked Not Active when: • The device or interface has not yet been started. • A failure has been encountered during the link or interface activation phase. (Such a failure produces an error message to the operator console, indicating the cause.)
DAD Pend	Duplicate Address Detection (DAD) for the link-local address is in progress on the IPv6 interface.

PortName
The name of the OSA-Express port. This is the value that was specified on the PORTNAME parameter on the INTERFACE statement. This field is significant only for IPAQENET and IPAQENET6 interfaces.

Datapath

The subchannel address that is associated with the TRLE definition. This value is one of the addresses that was specified on the DATAPATH parameter on the TRLE definition and is the subchannel address that VTAM assigned to this interface. If VTAM has not yet assigned a subchannel address to this interface, then this field contains the value `Unknown`. This field is significant only for IPAQENET, IPAQIDIO, IPAQENET6, and IPAQIDIO6 interfaces.

DatapathStatus

The datapath status. This field is significant only for IPAQENET, IPAQIDIO, IPAQENET6 , and IPAQIDIO6 interfaces. This field contains information that is useful if the interface is not activating correctly. See Table 16 on page 445 for possible status values.

CHPIDType

The CHPID type that is associated with this interface. This value was specified on the CHPIDTYPE parameter on the INTERFACE statement (or was generated by the stack) for OSA-Express QDIO interfaces. This field is significant only for IPAQENET and IPAQENET6 interfaces. The possible values and meanings are:

OSD A CHPID with connectivity to the external data network

OSX A CHPID with connectivity to the intra ensemble data network

OSM A CHPID with connectivity to the intra node management network

IPAddr

The IP address and optional number of bits (leftmost significant bits), which identifies the subnet mask of the interface. This value was specified on the IPADDR parameter on the INTERFACE statement. This field is significant for IPAQENET interfaces only. If the interface is defined with the TEMPIP keyword, the IP address is 0.0.0.0.

CHPID

The CHPID value that is associated with this interface. For HiperSockets, this value was specified on the CHPID parameter on the INTERFACE statement for predefined HiperSockets interfaces or is the value obtained from VTAM for HiperSockets interfaces that are created by dynamic XCF definitions. For OSA-Express QDIO interfaces that are configured with CHPIDTYPE OSX, this value was specified on the CHPID parameter. This field is significant only for IPAQIDIO6, IPAQENET, or IPAQENET6 interfaces.

SMCR

Indicates whether this interface can be used for new TCP connections for Shared Memory Communications over Remote Direct Memory Access (SMC-R) for external data network communications. This value was specified on the SMCR or NOSMCR parameter on the INTERFACE statement for OSA-Express QDIO interfaces. This field is significant only for IPAQENET and IPAQENET6 interfaces. The possible values and meanings are:

YES Indicates that this interface can be used for new TCP connections to communicate with other stacks on the external data network by using SMC-R.

For an inactive interface, Yes means the interface is configured for SMC-R. An interface is configured for SMC-R when the SMCR parameter was specified on the INTERFACE statement or is in effect by default.

For an active interface, Yes means the interface is enabled for SMC-R. An interface is enabled for SMCR when the following conditions are true:

- The SMCR parameter was specified on the INTERFACE statement or is in effect by default.
- The TCP/IP stack is enabled for SMC-R. A TCP/IP stack is enabled for SMC-R when the SMCR parameter was specified on the GLOBALCONFIG statement.
- A physical network ID value was configured in HCD for this interface.

NO Indicates that this interface cannot be used for new TCP connections to communicate with other stacks on the external data network by using SMC-R. The NOSMCR parameter was specified on the INTERFACE statement.

Disabled (*reason_text*)
Indicates that this interface was configured to communicate with other stacks on the external data network by using SMC-R, but SMC-R cannot be used for new TCP connections because of one of the following reasons:

No PNetID
No physical network ID value was configured in HCD for this interface. The physical network ID is learned during interface activation so this reason text is only valid for an active interface.

GLOBALCONFIG NOSMCR
The TCP/IP stack was not enabled for SMC-R.

No Subnet Mask
No subnet mask was configured on the INTERFACE statement for this interface.

PFID The Peripheral Component Interconnect Express (PCIe) function ID (PFID) value that defines an 10GbE RoCE Express feature. This value is specified on the SMCR PFID parameter of the GLOBALCONFIG TCP/IP profile statement. This field is significant only for RNIC interfaces that are created when an IPAQENET or IPAQENET6 interface specifies SMCR or takes SMCR as the default setting.

PortNum
Specifies the 10GbE RoCE Express port number that is used for the associated PFID. The PortNum value is specified with the PFID value on the SMCR parameter of the GLOBALCONFIG statement in the TCP/IP profile.

PNetID
The physical network ID value that is configured in HCD for an interface. This field is significant only for IPAQENET interfaces defined by using the INTERFACE statement, IPAQENET6 interfaces, and active RNIC interfaces.

Interface	Value
Active OSD interfaces	• If a physical network ID is configured in HCD for the OSD interface, the configured value is displayed. • If no physical network ID is configured in HCD for the OSD interface, the value*None* is displayed. If the OSD interface is configured to use SMCR, a value of Disabled (No PNetID) is displayed in the SMCR field.
Active OSX interfaces	The reserved value IEDN is used.
Active RNIC interfaces	The value that is configured in HCD for the RNIC interface is displayed. If no value is configured in HCD, activation of the RNIC interface fails.

TRLE The name of the TRLE that is associated with this interface. This field is significant only for MPCPTP6, IPAQIDIO, IPAQIDIO6 and RNIC interfaces.

For MPCPTP6 interfaces
This value was specified on the TRLE parameter of the INTERFACE statement for predefined MPC interfaces or is the value obtained from VTAM for MPC interfaces that are created by dynamic XCF definitions.

For IPAQIDIO or IPAQIDIO6 interfaces
This value is obtained from VTAM for IPAQIDIO or IPAQIDIO6 interfaces that INTERFACE definitions create. This value is displayed for active interfaces only.

For RNIC interfaces
This value is obtained from VTAM for RNIC interfaces that are created for PFIDs configured on the GLOBALCONFIG statement when SMC-R is enabled. This value is displayed only when the PFIDStatus value of the interface is Starting or Ready.

PFIDStatus
This field is the RNIC interface PFID status. The following list describes several status values:

PFID status	Description
Ready	The initialization sequence with the PFID is complete. The PFID is ready.
Not Active	The PFID is not active. The PFID has never been started, or has been stopped after having been started.
Starting	A START command of the PFID has been issued, and TCP/IP has sent an Activation request to the Data Link Control (DLC) layer.
Deactivating	DLC has performed the first stage of an orderly PFID deactivation.

GidAddr
The group identifier (GID) value that is associated with the RNIC interface. This value is obtained from VTAM for RNIC interfaces that are created for

PFIDs configured on the GLOBALCONFIG statement when SMC-R is enabled. This value is displayed for active RNIC interfaces only.

NetNum
The adapter number that was specified on the LINK statement. This field is significant only for CTC and LCS links.

QueSize
The queue size represents the number of outbound packets for this link or interface that are queued and waiting for ARP or neighbor resolution. This field is significant only for links on ATM and LCS devices and for IPAQENET6 interfaces.

Speed Indicates the interface speed (in million bits per second) that is reported by the device. This field is significant only for IPAQENET links or interfaces, ATM and IPAQTR links, and IPAQENET6 interfaces, and only if the link or interface is active.

MAC address order (MacAddrOrder)
Indicates the canonical option (CANON/NONCANON) that is specified on the LINK statement. This field is significant only for token-ring links.

SrBridgingCapability
Indicates whether the link supports source route bridging. This field is significant only for token-ring links.

IpBroadcastCapability
Indicates whether the link is broadcast capable. This field is significant only for links on LCS and MPCIPA devices and IPAQENET interfaces.

ArpBroadcastType
Indicates the ARP broadcast option (ALLRINGSBCAST/LOCALBCAST) that is specified on the LINK statement. This field is significant only for token-ring links.

ArpOffload
Indicates whether ARP processing is being offloaded to the adapter. This field is significant only for active links that support ARP offload.

ArpOffloadInfo
Indicates whether the adapter reports ARP offload data to TCP/IP. If so, then the ARP cache data can be displayed with the Netstat ARP/**-R** report even though the ARP function is being offloaded. This field is significant only for active links that support ARP offload.

Routing Parameters
This section displays routing information for IPv4 links that are defined with the DEVICE and LINK profile statements.

MTU Size
This value is determined in one of the following ways:

- If you are using OMPROUTE and the link is defined to OMPROUTE, the value might have been specified on the MTU parameter on the OSPF_INTERFACE, RIP_INTERFACE, or INTERFACE statement for the link. If one of these OMPROUTE statements was specified for the link but the MTU parameter was not specified, OMPROUTE sets the **MTU Size** value to 576.
- If you are using OMPROUTE, the link is not defined to OMPROUTE, and OMPROUTE is not configured to ignore undefined links, OMPROUTE sets the **MTU Size** value to 576.

- If you are not using OMPROUTE (or if the link is not defined to OMPROUTE), OMPROUTE is configured to ignore undefined links, and a BSDROUTINGPARMS profile statement was specified for the link, then the **MTU Size** value is configured using the BSDROUTINGPARMS profile statement MTU parameter.
- If none of the previously described methods provides an MTU Size value or if the MTU Size parameter does not apply to this link, then the value `n/a` is displayed.

To determine the MTU Size value that is being used by the stack for a link, see the ActMtu field for the link. To determine the MTU Size value that is being used for a route over this link, see the MTU field on the Netstat ROUTe/**-r** report.

Metric The routing metric that is associated with the link. This value is determined in one of the following ways:

- If you use OMPROUTE and the link is defined to OMPROUTE using the OSPF_INTERFACE statement, then the Metric value is configured using the Cost0 parameter on the OSPF_INTERFACE statement. If the Cost0 parameter is not specified, then OMPROUTE sets the value to 1.
- If you use OMPROUTE and the link is defined to OMPROUTE using the RIP_INTERFACE statement, then the Metric value is configured using the In_Metric parameter on the RIP_INTERFACE statement. If the In_Metric parameter is not specified, then OMPROUTE sets the value to 1.
- If you use OMPROUTE and the link is defined to OMPROUTE using the INTERFACE statement or if the link is not defined to OMPROUTE and OMPROUTE is not configured to ignore undefined links, then OMPROUTE sets the Metric value to 0.
- If you are not using OMPROUTE (or if the link is not defined to OMPROUTE) and OMPROUTE is configured to ignore undefined links, the Metric value is configured in one of the following ways:
  - For dynamic XCF links, the Metric value is configured using the cost_metric value of the DYNAMICXCF parameter on the IPCONFIG profile statement.
  - If a BSDROUTINGPARMS profile statement was specified for the link, the Metric value is configured using the cost_metric parameter of BSDROUTINGPARMS profile statement.
- If none of the previously described methods provided a Metric value, the stack sets the value to 0

DestAddr

The destination address applies to point-to-point links only and is the IP Address of the other side of the point-to-point link. This value is determined in one of the following ways:

- If you are using OMPROUTE and the link is defined to OMPROUTE, then the value is configured using the Destination_Addr parameter on the OSPF_INTERFACE, RIP_INTERFACE, or INTERFACE statement. If the Destination_Addr parameter is not specified, then OMPROUTE sets the value to 0.

- If you are using OMPROUTE but the link is not defined to OMPROUTE and OMPROUTE is not configured to ignore undefined links, then OMPROUTE sets the value to 0.
- If you are not using OMPROUTE (or if the link is not defined to OMPROUTE), OMPROUTE is configured to ignore undefined links, and a BSDROUTINGPARMS profile statement was specified for the link, then the value is configured using the dest_addr parameter for this statement.
- If none of these methods has provided a destination address value, then the stack sets a default value in one of the following ways:
  - For links other than point-to-point links, the value is set to 0.
  - For point-to-point links, the value is set as follows:
    - If routes are defined over the link, then the stack sets the value using the gateway address of an indirect route or the destination address of a direct host route.
    - If no routes are defined over the link, then the value is set to 0.

SubnetMask

The subnet mask that is associated with the link. This value is determined in one of the following ways:

- If you are using OMPROUTE and the link is defined to OMPROUTE, then the value is configured using the Subnet_Mask parameter on the OSPF_INTERFACE, RIP_INTERFACE, or INTERFACE statement.
- If you are using OMPROUTE, the link is not defined to OMPROUTE, and OMPROUTE is not configured to ignore undefined links, then OMPROUTE assigns a value based on the IP address that is assigned to the link.
- If you are not using OMPROUTE (or if the link is not defined to OMPROUTE) and OMPROUTE is configured to ignore undefined links, then the value is assigned in one of the following ways:
  - For dynamic XCF links, the value is configured using the *subnet_mask* or *num_mask_bits* value of the DYNAMICXCF parameter on the IPCONFIG profile statement.
  - For dynamic VIPA links, the value is configured using the *address_mask* parameter on the VIPADEFINE, VIPABACKUP, or the VIPARANGE profile statement.
  - If a BSDROUTINGPARMS profile statement was specified for the link, the value is configured using the *subnet_mask* parameter for the BSDROUTINGPARMS profile statement.
- If none of the previously described methods provides a subnet mask value, then the stack assigns a value based on the IP address that is assigned to the link.

Packet trace settings

Use the PKTTRACE statement to control the packet tracing facility in TCP/IP. You can use this statement to select IP packets as candidates for tracing and subsequent analysis. An IP packet must meet all of the conditions specified on the statement for it to be traced.

Protocol
The protocol number from the PROT keyword of the PKTTRACE command or * if not specified.

TrRecCnt
The number of packets traced for this PKTTRACE command.

PckLength
The value of the ABBREV keyword of the PKTTRACE command or FULL to capture the entire packet.

SrcPort
The port number from the SRCPORT parameter of the PKTTRACE command or profile statement. If an asterisk (*) is displayed, then either a port number was not specified for the SRCPORT parameter, or the PORTNUM parameter was also specified. If both the SrcPort and PortNum fields contain a value *, then the IP packets are not being filtered by the source port.

DestPort
The port number from the DESTPORT parameter of the PKTTRACE command or profile statement. If an asterisk (*) is displayed, then either a port number was not specified for the DESTPORT parameter, or the PORTNUM parameter was also specified. If both the DestPort and PortNum fields contain an asterisk (*), then the IP packets are not being filtered by destination port.

PortNum
The port number from the PORTNUM parameter of the PKTTRACE command or profile statement. If an asterisk (*) is displayed, then either a port number was not specified for the PORTNUM parameter, or the DESTPORT or SRCPORT parameters were also specified. If the PortNum, SrcPort, and DestPort fields all contain an asterisk (*), then the IP packets are not being filtered by port.

Discard
The value specified for the PKTTRACE DISCARD parameter. A numerical value is a discard reason code. The value NONE, which is the default, indicates that only packets that were delivered are being traced. The value ALL indicates that only discarded IP packets are being traced. The value asterisk (*) indicates that discarded IP packets and delivered IP packets are being traced.

IpAddr
The IP address from the IPADDR keyword of the PKTTRACE command or asterisk (*) if not specified.

SubNet
The IP subnet mask from the SUBNET keyword of the PKTTRACE command or asterisk (*) if not specified.

ATM Specific
This section contains information about ATM links:

ATM PortName
The PORTNAME value specified on the DEVICE statement.

For an ATM link configured as a Permanent Virtual Circuit (PVC), the following additional fields are displayed:

ATM PVC Name
The name of the PVC specified on the ATMPVC statement.

PVC Status
This field can have the following values:

ATM PVC status	Description
Not Active	The PVC is not active. There is no command to start a PVC; PVC activation is normally attempted during START device processing. A PVC is marked Not Active when: • The device has not yet been started. • The remote side of the PVC is not active. • A failure has been encountered during the PVC activation phase. (Such a failure produces an error message to the operator.)
Ready	The initialization sequence for the PVC is complete. The PVC is now ready for use.

For an ATM link configured as a Switched Virtual Circuit (SVC), the following additional fields are displayed:

ATM LIS Name
The name of the ATM Logical IP Subnet (LIS) specified on the ATMLIS statement.

SubnetValue
The subnet_value specified on the ATMLIS statement.

SubnetMask
The subnet_mask specified on the ATMLIS statement.

DefaultMTU
The DFLTMTU value specified on the ATMLIS statement.

InactvTimeOut
The INACTVTO value specified on the ATMLIS statement.

MinHoldTime
The MINHOLD value specified on the ATMLIS statement.

MaxCalls
The maximum number of SVCs that can be active for this ATMLIS.

CachEntryAge
The CEAGE value specified on the ATMLIS statement.

ATMArpReTry
The ARPRETRIES value specified on the ATMLIS statement.

ATMArpTimeOut
The ARPTO value specified on the ATMLIS statement.

PeakCellRate
The PEAKCR value specified on the ATMLIS statement.

NumOfSVCs
The number of currently active SVCs for this ATMLIS.

BearerClass
The BEARERCLASS value specified on the ATMLIS statement.

For an ATM SVC link that is configured with an ATM ARP server, the following additional fields are displayed:

ATMARPSV Name

The name of the ATM ARP server specified on the ATMARPSV statement.

VcType

Indicates whether the ATM ARP server connection is a PVC or an SVC. This value comes from the ATMARPSV statement.

ATMaddrType

The ATM address type specified on the ATMARPSV statement. The only supported value is NSAP.

ATMaddr

The ATM address of the ATM ARP server. If the connection to the ATM ARP server is an SVC, then this is the physical_addr value specified on the ATMARPSV statement. For a PVC connection to the ATM ARP server, this is the remote ATM address learned by TCP/IP when the PVC was activated.

IpAddr

The IP address of the ATM ARP server. If the connection to the ATM ARP server is an SVC, then this is the ip_addr value specified on the ATMARPSV statement. For a PVC connection to the ATM ARP server, this is the remote IP address learned by TCP/IP when the PVC was activated.

Multicast Specific

This section displays multicast information for the link or interface.

Multicast Capability

Indicates whether the link or interface is multicast capable.

- For point-to-point interfaces, the value of this field is always `Yes`.
- For LCS and MPCIPA links and IPAQENET, IPAQENET6, IPAQIDIO, and IPAQIDIO6 interfaces, the multicast capability is known only after the link or interface is active. If the link or interface is not active, the multicast capability value is `Unknown`.
- For IPAQIQDX6 interfaces, the value of this field is always `ND only`, the interface is multicast capable but multicast processing is used only for neighbor discovery.

If the link or interface is multicast capable then the following additional fields are displayed for each multicast group for which the link or interface is receiving data. There is no limit to the number of multicast groups for which a link or interface can receive data. For IPAQIQDX6 interfaces, the multicast groups indicate only neighbor discovery processing.

Group The multicast group address for which this link or interface is receiving data.

RefCnt

The number of applications that are receiving data for this multicast group.

SrcFltMd

The source filter mode indicates the type of multicast source IP address filtering that has been configured at the interface. Source IP address filtering can be done by either an IGMPv3 or MLDv2-capable multicast router on a per interface basis or by the host on a per socket basis. The host provides its source filter mode and source IP address filter list for each multicast group that an application has joined on the interface with any IGMPv3 and MLDv2-capable multicast routers that are connected to the interface. This permits IGMPv3-capable and MLDv2-capable multicast routers to send only multicast packets that have been requested by at least one host on the subnet to which the interface is connected. If the multicast packets are not filtered by an IGMPv3-capable or MLDv2-capable multicast router (for example the router does not support IGMPv3 or MLDv2), or if there are multiple hosts on the local area network that have either a different source filter mode or a different source IP address filter list for a given multicast group, the host uses the source IP address filter information to ensure that each application receives only packets that it has requested.

The value is either Include or Exclude. A source filter applies only to incoming multicast data. The source filter applies to all the IP addresses in the SrcAddr fields for the associated multicast group address and the link or the interface. The source filter mode and the corresponding source filter IP addresses are configured by applications for their UDP or RAW sockets that have joined the multicast group for this interface. See the information about Designing multicast programs in the z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference for details about how applications configure these values for a socket.

Include
Indicates that the interface or link receives only multicast datagrams that have a source IP address that matches an IP address indicated in the SrcAddr field.

Exclude
Indicates either that the source filter function is not active or that the interface or link receives only multicast datagrams that have a source IP address that does not match an IP address indicated in the SrcAddr field. If the source filter function is not active or if the source filter function is active but no SrcAddr value is set, the SrcAddr field contains the value None.

SrcAddr
Source address information for the socket.

ipaddr The source IP address that is used in conjunction with the SrcFltMd value to determine which incoming multicast datagrams are received by the interface.

None This value is displayed only when the source filter function is not configured for the interface or when the source filter mode is Exclude but there was no intersection of excluded source IP addresses among the sockets for the same multicast group address and interface.

Source VIPA interface (SrcVipaIntf)
The name of the VIPA that is used for this interface if source VIPA is in effect. This is the value that was specified on the

SOURCEVIPAINTERFACE parameter on the INTERFACE statement. This field is significant only for IPAQENET, IPAQENET6, IPAQIDIO6, and MPCPTP6 interfaces.

Duplicate address detection (DupAddrDet)
The DUPADDRDET value specified on the INTERFACE statement. This field is significant only for IPAQENET6 interfaces.

Interface ID (IntfID)
The INTFID value specified on the INTERFACE statement. This field is significant only for IPAQENET6, IPAQIDIO6, and MPCPTP6 interfaces.

MAC address (MacAddress)
The local hardware address for this link or interface. This field is significant only for links on LCS devices and for IPAQENET6 interfaces. This field is displayed only if the link or interface is active and if virtual MAC (VMAC) is not configured.

Router Hop Limit (RtrHopLimit)
The value that is placed in the Hop Count field of the IP header for outgoing IP packets. This value was obtained from a received router advertisement and is significant only for IPAQENET6 interfaces. This field is displayed only when a nonzero hop limit value was received in a router advertisement over this interface and IGNOREROUTERHOPLIMIT is not configured on the IPCONFIG6 profile statement.

CfgMtu
The MTU value that was configured on the INTERFACE statement (or `None` if an MTU value was not configured). This field is significant only for IPAQENET, IPAQENET6, or IPAQIDIO interfaces.

ActMtu
The largest MTU that is supported by an active link or interface. If the link or interface is inactive, then this field displays Unknown. This field is significant for all links and interfaces except virtual ones.

VLANid
This field is significant only for IPAQENET links or interfaces, IPAQIDIO links, or IPAQENET6 and IPAQIDIO6 interfaces. This field indicates whether a virtual LAN ID was configured on the VLANID parameter on the LINK or INTERFACE profile statement. The following values can be displayed in this field:

`None`
This value indicates that the VLANID parameter was not specified on the LINK or INTERFACE profile statement for the interface. For an IPAQIDIO link or IPAQIDIO6 interface that is dynamically generated as part of dynamic XCF HiperSockets processing, this value indicates that the IQDVLANID parameter was not specified on the GLOBALCONFIG profile statement.

`n/a`
This value indicates that the VLANID parameter was specified on the LINK or INTERFACE profile statement, but the interface does not support VLAN IDs.

`vlanid`
If an OSA-Express device is active and supports virtual LAN IDs, this field indicates that all IP packets through this OSA-Express link or interface from this stack are being tagged with this VLAN ID. For an active HiperSockets link or interface that supports virtual LAN IDs,

this field indicates that all IP packets through this HiperSockets link or interface from this stack are associated with this VLAN ID.

VLANpriority

This field is significant only for active IPAQENET links or interfaces or IPAQENET6 interfaces. This field indicates whether all IP packets through this OSA-Express link or interface from this stack are being tagged with a VLAN priority. The possible values are:

Enabled

Indicates that all IP packets through this OSA-Express link or interface are being tagged with a VLAN priority. See z/OS Communications Server: IP Configuration Reference for information about the SetSubnetPrioTosMask statement and details about how to configure VLAN priorities.

Disabled

Indicates that the OSA-Express link or interface supports VLAN priority, but currently no VLAN priority values are defined. If the VLANid field displays `None` or `n/a`, all IP packets through this OSA-Express link or interface are not VLAN tagged. All other values indicate that all IP packets are VLAN tagged, but only with VLAN IDs, not with VLAN priority.

Unknown

Indicates that the VLAN priority tagging support for the OSA-Express is unknown because the link or interface is not yet active.

DynVLANRegCfg

This field is significant only for IPAQENET links or interfaces and IPAQENET6 interfaces. This field is displayed only under the following conditions:

- The link or interface is not yet active and a VLAN ID was specified.
- The link or interface is active, a VLAN ID value was specified, and the OSA-Express feature has accepted the VLAN ID value.

This field indicates whether dynamic VLAN ID registration was configured on the LINK or INTERFACE statement. The possible values are:

Yes

Indicates that the DYNVLANREG parameter was specified on the LINK or INTERFACE statement.

No Indicates that the NODYNVLANREG parameter was specified on the LINK or INTERFACE statement or is in effect by default.

DynVLANRegCap

This field indicates whether the OSA-Express feature that is represented by the LINK or INTERFACE statement is capable of supporting dynamic VLAN ID registration.This field is significant only for IPAQENET links or interfaces and IPAQENET6 interfaces. This field is displayed only under the following conditions:

- The link or interface is not yet active and a VLAN ID was specified.
- The link or interface is active, a VLAN ID value was specified, and the OSA-Express feature has accepted the VLAN ID value.

The possible values are:

Yes
Indicates that the OSA-Express feature is capable of supporting dynamic VLAN ID registration.

No Indicates that the OSA-Express feature is not capable of supporting dynamic VLAN ID registration.

Unknown
Indicates that the dynamic VLAN ID registration capability of the OSA-Express feature is unknown because the link or interface is not yet active.

ChecksumOffload
This field is significant only for active IPAQENET and IPAQENET6 links or interfaces. This field indicates whether the checksum offload support is in effect and is displayed only when the link or interface is active. The possible values are:

Yes Indicates that the checksum offload function is enabled on the adapter for this interface.

No Indicates that the checksum offload function is not enabled on the adapter for this interface.

Unsupported
Indicates that the checksum offload function is not supported on the adapter for this interface.

SegmentationOffload
This field is significant only for active IPAQENET and IPAQENET6 links or interfaces. This field indicates whether the TCP segmentation offload support is in effect and is displayed only when the link or interface is active. Possible values are:

Yes Indicates that the segmentation offload function is enabled on the adapter for this interface.

No Indicates that the segmentation offload function is not enabled on the adapter for this interface.

Unsupported
Indicates that the segmentation offload function is not supported on the adapter for this interface.

SecClass
This field identifies the security class value for IP filtering. This field applies to all IPv4 and IPv6 interfaces except virtual and loopback, but the value is in effect only if the IPSec function is active for the applicable IP version. You can use the Netstat CONFIG/**-f** command to determine whether IPSec is active. Valid security class values are in the range 1 - 255. The displayed value was defined by one of the following methods:

- By the SECCLASS parameter on the LINK or INTERFACE profile statement
- For dynamic XCF interfaces, by the DYNAMICXCF SECCLASS subparameter on the IPCONFIG or IPCONFIG6 profile statement
- For OSM interfaces, by the TCP/IP stack's automatic configuration of the interface, or by the IPSECURITY OSMSECCLASS subparameter on the IPCONFIG6 profile statement

MonSysplex
Indicates whether the status of this link or interface is being monitored by

Sysplex Autonomics. This field is significant for all IPv4 links or interfaces except virtual, loopback, and all dynamically configured links, and for all IPv6 interfaces except virtual, loopback, and all dynamically configured interfaces.

Yes Indicates that the status of this link or interface is being monitored by Sysplex Autonomics. It is configured by specifying the MONSYSPLEX keyword on the LINK or INTERFACE profile statement and specifying the MONINTERFACE keyword for the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement. If DYNROUTE keyword is also coded on the GLOBALCONFIG SYSPLEXMONITOR profile statement, then the presence of dynamic routes over this link or interface is also monitored.

Configured
Indicates that this link or interface was configured to be monitored by Sysplex Autonomics. It was configured by specifying the MONSYSPLEX keyword on the LINK or INTERFACE profile statement, but the link or interface is not currently being monitored because the MONINTERFACE keyword was not specified on the SYSPLEXMONITOR parameter on the GLOBALCONFIG profile statement.

No Indicates that the status of this link or interface is not being monitored by Sysplex Autonomics because the MONSYSPLEX keyword was not specified on the LINK or INTERFACE profile statement.

Isolate
This field is significant only for IPAQENET interfaces (defined using the INTERFACE statement) and for IPAQENET6 interfaces. This field ndicates whether the OSA-Express device is prevented from routing packets directly to another stack that is sharing the OSA-Express connection. For more details, see OSA-Express connection isolation information in z/OS Communications Server: IP Configuration Guide.

No Indicates that this interface is eligible for OSA-Express direct routing. Therefore, the OSA-Express device can route packets directly to another stack that is sharing the OSA-Express connection (as long as the interface from the other stack is also eligible for direct routing).

Yes Indicates that this interface is not eligible for OSA-Express direct routing. Therefore, the OSA-Express device cannot routing packets directly to another stack that is sharing the OSA-Express connection.

OptLatencyMode
This field is significant only for IPAQENET interfaces (defined using the INTERFACE statement) and for IPAQENET6 interfaces. The field indicates whether optimized latency mode (OLM) was configured for this OSA-Express interface. For more information about optimized latency mode, see optimized latency mode information in z/OS Communications Server: IP Configuration Guide. Possible values are:

`No` Indicates that the OSA-Express interface is not configured with optimized latency mode.

Yes
Indicates that the OSA-Express interface is configured with optimized latency mode. Optimized latency mode optimizes interrupt processing for both inbound and outbound data.

Disabled
Indicates that the OSA-Express interface was configured with optimized latency mode, but the function could not be enabled when the interface was activated. The most likely reason is that the OSA-Express interface does not support this function.

IQDMultiWrite
This field is significant only for active HiperSockets devices or interfaces. This field indicates whether the HiperSockets multiple write facility is currently being used for the device or interface. To configure the stack to use the HiperSockets multiple write facility, specify the IQDMULTIWRITE parameter on the GLOBALCONFIG profile statement. The possible values are:

Enabled
Indicates that the HiperSockets multiple write facility is currently being used for the device or interface.

Enabled (ZIIP)
Indicates that the HiperSockets multiple write facility is currently being used for the device or interface. Additionally, CPU cycles that are associated with the HiperSockets multiple write facility are to be displaced to an available zIIP.

Disabled
Indicates that the HiperSockets multiple write facility is not currently being used for the device or interface.

Unsupported
Indicates that the IBM System z environment does not support the HiperSockets multiple write facility.

ReadStorage
This field is significant only for active IPAQENET and IPAQIDIO links or interfaces, IPAQTR links, and for IPAQIDIO6 and IPAQENET6 interfaces. This field indicates the amount of storage (in kilobytes) that is being used for read processing.

InbPerf
This field is significant only for IPAQENET links or interfaces, IPAQTR links, and IPAQENET6 interfaces. This field indicates how frequently the adapter interrupts the host. This field indicates how the processing of inbound traffic is performed. If the interface is not active, then this field shows the configured value. If the interface is active, then this field shows the value that is in effect. The possible values are:

Balanced
Indicates that the adapter is to use a static interrupt-timing value that strikes a balance between MinCPU and MinLatency.

Dynamic
This setting is significant only for IPAQENET links, and IPAQENET and IPAQENET6 interfaces. It indicates that the stack and the adapter are to dynamically update the frequency with which the adapter interrupts the host for inbound traffic.

WorkloadQueueing

This field is displayed only for IPAQENET and IPAQENET6 interfaces. It indicates whether QDIO inbound workload queueing is enabled. Possible values are:

Yes QDIO inbound workload queueing is in effect. The QDIO interface is defined using the INTERFACE statement with INBPERF DYNAMIC WORKLOADQ specified.

No QDIO inbound workload queueing is not in effect. The QDIO interface is defined using the INTERFACE statement with INBPERF DYNAMIC or INBPERF DYNAMIC NOWORKLOADQ specified.

Unsupported

QDIO inbound workload queueing was requested on the INTERFACE statement but the OSA-Express interface does not support it. QDIO inbound workload queueing is supported on OSA-Express3 or later features on an IBM System z10™ GA3 or later CPC.

MinCPU

Indicates that the adapter is to use a static interrupt-timing value that minimizes host interrupts, and therefore minimizes host CPU consumption.

MinLatency

Indicates that the adapter is to use a static interrupt-timing value that minimizes latency delay by more aggressively presenting received packets to the host.

TempPrefix

This field is significant only for IPAQENET6 interfaces with stateless address autoconfiguration enabled. One or more TempPrefix fields are displayed. Together the TempPrefix fields indicate the set of prefixes for which temporary IPv6 addresses can be generated, if temporary addresses are enabled on the IPCONFIG6 statement. The set of prefixes is specified on the TEMPPREFIX parameter on the INTERFACE statement. The possible values displayed are:

All IPv6 temporary addresses are generated for all prefixes that are learned from a router advertisement over this interface. This is the default.

Disabled

Autoconfiguration of temporary addresses for the interface is disabled because duplicate addresses were detected. Temporary addresses are not generated for this interface.

None Temporary addresses are not generated for this interface.

IPv6 prefix/prefix length

IPv6 temporary addresses are generated for all prefixes that are learned from a router advertisement over this interface and that are included in one of the prefixes in this prefix list.

Link/Interface Statistics

This section is significant for all links and interfaces except virtual ones. The following statistical information is displayed:

BytesIn

Number of bytes received over an interface.

Inbound Packets

The number of unicast inbound packets received over an interface. This value applies to all links and interfaces except for RNIC interfaces.

Inbound Packets In Error

Number of inbound packets discarded because of an error validating the packet. This value applies to all links and interfaces except for RNIC interfaces.

Inbound Packets Discarded

Number of inbound packets discarded because of an out-of-storage condition. This value applies to all links and interfaces except for RNIC interfaces.

Inbound Packets With No Protocol

Number of inbound packets discarded because of an unknown protocol type. This value applies to all links and interfaces except for RNIC interfaces.

BytesOut

Number of bytes transmitted over an interface.

Outbound Packets

The number of unicast outbound packets transmitted over an interface. This value applies to all links and interfaces except for RNIC interfaces.

Outbound Packets In Error

Number of outbound packets discarded because of errors other than an out-of-storage condition. This value applies to all links and interfaces except for RNIC interfaces.

Outbound Packets Discarded

Number of outbound packets discarded because of an out-of-storage condition. This value applies to all links and interfaces except for RNIC interfaces.

Inbound Operations

Number of Remote Direct Memory Access (RDMA) inbound operations processed across this interface. This value applies to RNIC interfaces only.

Outbound Operations

Number of RDMA outbound operations processed across this interface. This value applies to RNIC interfaces only.

SMC Links

Current number of SMC-R links between this stack and other stacks across this interface. This value applies to RNIC interfaces only.

TCP Connections

Number of TCP connections across all the SMC-R links that are associated with this interface. One or more TCP connections can use the same SMC-R link. This value applies to RNIC interfaces only.

Intf Receive Buffer Inuse
Amount of RMB storage in use by the TCP connections that are using the SMC-R links associated with this interface. This value applies to RNIC interfaces only.

IPv4 LAN Group Summary
The IPv4 LAN group summary lists links or interfaces that are takeover candidates for each other. The stack creates a LAN group when it detects redundant connectivity to a LAN. For each link or interface in the LAN group, this summary displays which link or interface owns ARP responsibility for that link or interface. The summary also displays which link or interface owns the ARP responsibility in the LAN group for any VIPAs.

IPv6 LAN Group Summary
The IPv6 LAN group summary lists interfaces that are takeover candidates for each other. The stack creates a LAN group when it detects redundant connectivity to a LAN. For each interface in the LAN group, this summary displays which interface owns neighbor discovery (ND) address resolution responsibility for that interface. The summary also displays which interface owns the ND Address Resolution responsibility in the LAN group for any VIPAs.

LanGroup
Identifies the LAN group. This identifier is assigned by the stack and represents a group of interfaces on the same LAN. This identifier is not a VLAN ID.

Name The link name configured on the LINK statement or the interface name configured on the INTERFACE statement.

Status The link or interface status. Valid values are Active or Not Active.

ArpOwner
The link or interface name that owns ARP responsibility for this link or interface in the LAN group. An active link or interface owns its ARP responsibility.

NDOwner
The interface name that owns neighbor discovery (ND) responsibility for this interface in the LAN group. An active interface owns its ND responsibility.

VipaOwner
Indicates whether the link or interface owns the ARP or ND responsibility for the VIPAs in the LAN group.

Associated IQDX Interface
The name of the Internal Queued Direct I/O extensions function (IQDX) interface that is associated with this OSX interface. This section is significant for OSX interfaces that use an IQDX interface for intraensemble data network (IEDN) connectivity. The following information is displayed:

IQDX Status
The status of the IQDX interface. See the description of the LnkStatus/IntfStatus field for the possible interface status values.

BytesIn
The number of bytes that have been received over the associated IQDX interface.

Inbound Packets
The number of unicast inbound packets that have been received over the associated IQDX interface.

BytesOut
The number of bytes that have been transmitted over the associated IQDX interface.

Outbound Packets
The number of unicast outbound packets that have been transmitted over the associated IQDX interface.

Associated RNIC Interface
The dynamic interface name that is generated for 10GbE RoCE Express interface that this stack uses for SMC-R communications. This field is significant only for active IPAQENET and IPAQENET6 interfaces that specify SMCR or take SMCR as the default value.

SMC Link Information
The SMC link information. This section is displayed for each RNIC interface only when the SMC modifier or the SMCID/-U filter is specified. The following fields and statistics are displayed.

Guideline: An SMC-R link is uniquely identified by the combination of the VLAN number, local GID, local VMAC address, local QP number, remote GID, remote VMAC address, and remote QP number.

LocalSMCLinkId
The SMC-R link identifier that this TCP/IP stack dynamically creates to represent the link.

RemoteSMCLinkId
The SMC-R link identifier that the remote peer uses to represent the link. The value is provided to this TCP/IP stack during link activation.

SMCLinkGroupId
The group identifier that this TCP/IP stack dynamically creates to represent the SMC-R link group that includes this individual link.

VLANid
The virtual LAN ID for this SMC-R link. The value None is displayed if a virtual LAN ID has not been configured.

MTU The negotiated MTU size that is used for this SMC-R link.

LocalGid
The local GID value that is associated with this SMC-R link. This is the same information that is displayed in the `GidAddr` field.

LocalMACAddr
The local virtual MAC address that is associated with this SMC-R link.

LocalQP
The local queue pair (QP) value that is associated with this SMC-R link.

RemoteGid
The peer GID value that is associated with this SMC-R link.

RemoteMACAddr
The peer virtual MAC address that is associated with this SMC-R link.

RemoteQP
The peer QP value that is associated with this SMC-R link.

SMCLinkBytesIn
Number of inbound data bytes transferred across this SMC-R link.

SMCLinkInOperations
Number of Remote Direct Memory Access (RDMA) inbound operations processed across this SMC-R link.

SMCLinkBytesOut
Number of outbound data bytes transferred across this SMC-R link.

SMCLinkOutOperations
Number of RDMA outbound operations processed across this SMC-R link.

TCP Connections
Number of TCP connections across this SMC-R link.

Link Receive Buffer Inuse
Amount of RMB storage in use by the active TCP connections that are associated with this SMC-R link.

32K Buffer Inuse
Amount of 32K RMB storage in use by the active TCP connections that are associated with this SMC-R link.

64K Buffer Inuse
Amount of 64K RMB storage in use by the active TCP connections that are associated with this SMC-R link.

128K Buffer Inuse
Amount of 128K RMB storage in use by the active TCP connections that are associated with this SMC-R link.

256K Buffer Inuse
Amount of 256K RMB storage in use by the active TCP connections that are associated with this SMC-R link.

Other Buffer Inuse
For RMB storage that is allocated as buffers larger than 256K, the amount of these other buffers that are in use by the active TCP connections that are associated with this SMC-R link. If no buffers larger than 256K are allocated, this information is not displayed.

Guidelines:

1. The LOOPBACK device and link are displayed. The LOOPBACK6 interface is displayed if the stack is enabled for IPv6.
2. The byte counts for number of bytes received and number of bytes transmitted are always 0 for VIPA links and interfaces.

3. If an MTU was configured on the INTERFACE statement, then the actual MTU is the minimum of the configured MTU and the physical MTU value supported by the interface.

Restrictions:

1. No link-related information, packet trace settings, or BSD parameters are displayed for a device that has no link defined.
2. The packet trace setting is displayed only when it is defined and set to ON.
3. ATM specific information is displayed only for ATM devices that have links defined.

OSA-Express Network Traffic Analyzer Information

This section displays all currently defined OSA interfaces that are dynamically created by VARY TCPIP,,OSAENTA commands or OSAENTA PROFILE statements.

OSA PortName

The port name value of the OSA that is currently defined for performing the OSA-Express network traffic analyzer (OSAENTA) function. This value was specified on the PORTNAME parameter of a VARY TCPIP,,OSAENTA command or on an OSAENTA PROFILE statement. The following information is specific to this *PortName* value.

OSA DevStatus

The device status. The following list shows the possible values:

Starting

An OSAENTA ON command or statement has been processed and TCP/IP has sent an activation request to the data link control (DLC) layer.

Sent SETUP

DLC has acknowledged the TCP/IP activation request and TCP/IP has requested that DLC perform the initial I/O sequence with the device.

Enabling

DLC has acknowledged the TCP/IP activation request and TCP/IP has requested that DLC allow data connections to be established for the device.

Connecting

DLC has accepted the initial I/O sequence request.

Negotiating

The initial I/O sequence with the device is complete and TCP/IP is performing additional link-layer initialization.

Ready The initialization sequence with the device is complete. The device is now ready.

Deactivating

DLC has performed the first stage of an orderly device deactivation.

Not Active

The device is not active. (The device has never been started or has been stopped after having been started.)

OSA IntfName
The name of the interface that is dynamically created to communicate with the OSA Express2 adapter.

OSA IntfStatus
The trace collection interface status. The following list shows the possible values:

Ready The OSA interface used for OSAENTA is accepting all trace requests from the host.

Not Active
The OSA interface that is used for OSAENTA is not active. Either trace collection is disabled or else an error occurred during activation of the OSA interface that is to be used for trace collection. Such an error condition generates an error message on the operator console.

OSA Speed
The speed reported by the interface (in millions of bits per second).

OSA Authorization
The value of the OSA HMC authorization parameter. Possible values are Disabled, Logical Partition, PORT, CHPID, or UNKNOWN. The value is set to UNKNOWN until the first OSAENTA ON command has completed.

Disabled
The OSA does not allow the NTA function to trace any frames for the OSA.

Logical Partition
The OSA allows the NTA function to trace frames only for the current logical partition.

PORT The OSA allows the NTA function to trace frames for all stacks that share this OSA port.

CHPID
The OSA allows the NTA function to trace frames for all stacks that share the OSA.

UNKNOWN
The NTA trace interface has not been activated.

OSAENTA Cumulative Trace Statistics
Statistics accumulated for all frames that have been traced since the OSAENTA interface was first activated. These values are not reset by the OSAENTA ON command or statement.

DataMegs
The number of bytes of trace data (in megabytes) that have been received.

Frames
The total number of frames that have been traced.

DataBytes
The number of bytes of trace data that have been received.

FramesDiscarded
The number of frames that were traced but that the OSA device was not able to either forward to a host image or

deliver outbound. These packets are available for formatting in the CTRACE SYSTCPOT component, but have not been delivered to any user.

FramesLost
The number of frames that could not be recorded by TCP/IP in the SYSTCPOT buffers.

OSAENTA Active Trace Statistics
Statistics that have accumulated since the OSAENTA ON command or statement was last issued.

DataMegs
The number of bytes of trace data (in megabytes) that have been collected.

Frames
The total number of frames that have been collected.

DataBytes
The number of bytes of trace data that have been collected.

FramesDiscarded
The number of frames that were collected but that the OSA device was not able to either forward to a host image or deliver outbound. These packets are available for formatting in the CTRACE SYSTCPOT component, but have not been delivered to any user.

FramesLost
The number of frames that were not collected by TCP/IP in the SYSTCPOT buffers.

TimeActive
The number of minutes that have elapsed since the last OSAENTA ON command or statement.

OSAENTA Trace Settings
The current trace settings that are in effect for this OSAENTA interface.

Status The current trace status. Possible values are:

ON Tracing is enabled.

OFF Tracing is disabled.

DataMegsLimit
The amount of data (in megabytes) to be collected before the trace is automatically stopped. This value was specified on the DATA parameter.

FramesLimit
The number of frames to be collected before the trace is automatically stopped. This value was specified on the FRAMES parameter.

TimeLimit
The amount of time (in minutes) that data is collected before the trace is automatically stopped. This value was specified on the TIME parameter.

Abbrev
The size limit for the frames (in bytes) that are to be traced.

This value was specified on the ABBREV parameter. This value can be modified to reflect the size limit set by the OSA.

Discard

Identifies which frames being discarded by the OSA-Express device are to be traced. This value was specified on the DISCARD parameter. Possible values are:

All All frames discarded by the OSA-Express device are traced.

Exception

Frames discarded by the OSA-Express device for exception conditions are traced.

None No discarded frames are traced.

list A list of from one to eight values, that indicate the type of discarded frames that are to be traced by the OSA-Express device. This list includes decimal discard codes and the keyword parameter EXCEPTION.

OSAENTA Trace Filters

The values of the current accumulated filter variables from OSAENTA commands or statements for this OSA. If a filter variable has not been specified using OSAENTA commands or statements, then an asterisk is shown.

Nofilter

The filtering behavior when all filters (DEVICEID, MAC, ETHTYPE, VLANID, IPADDR, PROTOCOL, and PORTNUM) have been cleared or are inactive. This behavior applies when no filters have been specified, if the CLEARFILTER parameter is specified, or when the current setting for every filter is an asterisk (*). This filtering behavior applies only to packets that were not discarded by the OSA-Express device. This value was specified on the NOFILTER parameter. Possible values are:

All All frames are traced.

None No frames are traced.

DeviceID

Up to eight hexadecimal device identifiers that are specified on the DEVICEID keyword of an OSAENTA command or statement. The value is an asterisk (*) if no device identifiers were specified.

Mac Up to eight hexadecimal MAC addresses that are specified on the MAC keyword of an OSAENTA command or statement. The value is an asterisk (*) if no MAC addresses were specified.

VLANid

Up to eight decimal VLAN identifiers that are specified on the VLANID keyword of an OSAENTA command or statement. The value is an asterisk (*) if no VLAN identifiers were specified.

ETHType

Up to eight hexadecimal Ethernet types that are specified on the ETHTYPE keyword of an OSAENTA command or statement. The value is an asterisk (*) if no Ethernet types were specified. The name of the Ethernet type filter is displayed for commonly used Ethernet types, such as ARP, IPv4, IPv6, and SNA.

IPAddr

Up to eight dotted decimal IPv4 IP addresses and up to eight colon hexadecimal IPv6 IP addresses that are specified on the IPADDR keyword of an OSAENTA command or statement. The value is an asterisk (*) if no IP addresses were specified.

Protocol

Up to eight decimal protocol identifiers that are specified on the PROTOCOL keyword of an OSAENTA command or statement. The value is an asterisk (*) if no protocol identifiers were specified. The name of the protocol filter is displayed for commonly used protocols, while the protocol number is displayed for all others.

PORTNum

Up to eight decimal port numbers that are specified on the PORTNUM keyword of an OSAENTA command or statement. The value is an asterisk (*) if no port numbers were specified.

SMC Link Group Information

The information of the SMC link group. This section is displayed for each RNIC interface only when the SMC modifier or the SMCID/-U filter is specified. The following fields are displayed:

SMCLinkGroupId

The group identifier that this TCP/IP stack dynamically creates to represent the SMC-R link group that includes this individual link.

PNetID

The physical network ID value that is configured in HCD for this SMC-R link group.

Redundancy

The recovery and load balancing capabilities of the link group. The following list shows the possible values:

Full The link group has redundant active SMC-R links. Both the local and remote stacks have full failover capability. The z/OS server performs load balancing of TCP connections across the SMC-R links that are members of the link group.

Partial (Single local internal path)

The link group has redundant active SMC-R links. Both the local and remote stacks have failover capability. The z/OS server performs load balancing of TCP connections across the SMC-R links that are members of the link group. However, the links on the local stack have the same internal path.

| **Partial (Single local PCHID, unique ports)**

| The link group has redundant active SMC-R links. Both the

local and remote stacks have failover capability. The z/OS server performs load balancing of TCP connections across the SMC-R links that are members of the link group. However, the links on the local stack have the same PCHID with unique ports.

Partial (Single local PCHID and port)

The link group has redundant active SMC-R links. Both the local and remote stacks have failover capability. The z/OS server performs load balancing of TCP connections across the SMC-R links that are members of the link group. However, the links on the local stack have the same PCHID and port.

Partial (Single local RNIC)

The link group has multiple active SMC-R links and the remote stack has full failover capability, but the local stack has no failover capability. The z/OS server does not perform load balancing of TCP connections.

Partial (Single remote RNIC)

The link group has multiple active SMC-R links and the local stack has full failover capability, but the remote stack has no failover capability. The z/OS server does not perform load balancing of TCP connections.

None (Single local and remote RNIC)

The link group has a single active SMC-R link. Neither the local stack nor the remote stack has failover capability. The z/OS server cannot perform load balancing of TCP connections.

Link Group Receive Buffer Total

Amount of remote memory buffer (RMB) storage that is assigned to this SMC-R link group.

32K Buffer Total

Amount of 32K RMB storage that is assigned to this SMC-R link group.

64K Buffer Total

Amount of 64K RMB storage that is assigned to this SMC-R link group.

128K Buffer Total

Amount of 128K RMB storage that is assigned to this SMC-R link group.

256K Buffer Total

Amount of 256K RMB storage that is assigned to this SMC-R link group.

Other Buffer Total

For RMB storage that is allocated as buffers larger than 256K, the amount of these other buffers that are assigned to this SMC-R link group. If no buffers larger than 256K are allocated, this information is not displayed.

LocalSMCLinkId

The link identifier this TCP/IP stack dynamically creates to represent the SMC-R link in this SMC-R link group.

RemoteSMCLinkId

The SMC-R link identifier that the remote peer uses to represent the link in this SMC-R link group. The value is provided to this TCP/IP stack during link activation.

Netstat DRop/-D command

You can terminate a specific TCP/IP socket endpoint using the Netstat DRop/**-D** command.

When a DRop command is issued against a socket endpoint, any outstanding or following socket calls that refer to the socket that is being dropped terminate with a negative return code.

The socket endpoint that you drop can be a listening TCP server socket endpoint, a fully connected TCP socket (either server or client connection endpoint), or a UDP socket endpoint. When you drop a TCP connection or UDP endpoint, the associated socket does not close. The application that owns the associated socket is responsible for closing the socket.

The DRop/**-D** command terminates the socket endpoint that is identified by the connection number *n*. You can determine the connection number from the Conn column in the Netstat COnn/**-c** or Netstat TELnet/**-t** display.

You can use this parameter only if the MVS.VARY.TCPIP.DROP security product resource is defined and the user ID that is associated with the DROP command is permitted to this resource.

Use the DRop/**-D** command to terminate an individual TCP connection when you do not want to terminate the server itself, but want only to drop an individual connection with that server.

Use the DROP/**-D** command to terminate old TCP connections if they prevent a server from being restarted. This is sometimes necessary when the server does not enable the SO_REUSEADDR socket option before binding to its well-known port.

If you want to terminate all socket activity from a specific sockets application, terminate the application using the appropriate mechanism that is provided by the application. The DRop/**-D** command can have unpredictable results when issued against a listening socket or UDP socket. Some applications might not handle the subsequent socket errors as expected.

Restriction: The Netstat DRop/-D command supports dropping only one TCP connection per command invocation. The VARY TCPIP,,DROP command provides the ability to drop all TCP connections that are associated with a server. See "VARY TCPIP,,DROP" on page 243 for more information.

TSO syntax:

```
►►──NETSTAT DRop──n──┬─────────────────┬──────────────────────────────────────►◄
                     └─-TCp──tcpname──┘
```

z/OS UNIX syntax:

```
►►──netstat -D──n──┬──────────────┬──────────────────────────────────►◄
                   └─ -p──tcpname─┘
```

TCp/-p *tcpname*
Executes the command against a specific TCP/IP address space. The *tcpname* is an 8-byte procedure name that is used to start the TCP/IP. When the S member.identifier method of starting TCP/IP is used, the value specified for identifier must be used as *tcpname*.

n
The connection number that is a unique number assigned by the TCP/IP stack to uniquely identify a socket entity.

Command syntax examples:

From TSO environment:

```
NETSTAT DROP n
Drop the connection n from the default TCP/IP stack.
NETSTAT DROP m TCP TCPCS6
Drop the connection m from TCPCS6 stack.
```

From UNIX shell environment:

```
   netstat -D n
   netstat -D m -p tcpcs6
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT CONN

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
User Id  Conn     Local Socket           Foreign Socket        State
-------  ----     ------------           --------------        -----
PORTMP3  00010035 0.0.0.0..2220          0.0.0.0..0            Listen
TSUSER1  00010020 0.0.0.0..1027          0.0.0.0..0            Listen
TSUERS2  00010043 127.0.0.1..1033        127.0.0.1..23         Establsh
PORTMP3  00021002 0.0.0.0..2221          *..*                  UDP

NETSTAT DROP 10035
Connection successfully dropped

NETSTAT CONN

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:39
User Id  Conn     Local Socket           Foreign Socket        State
-------  ----     ------------           --------------        -----
TSUSER1  00010020 0.0.0.0..1027          0.0.0.0..0            Listen
TSUERS2  00010043 127.0.0.1..1033        127.0.0.1..23         Establsh
PORTMP3  00021002 0.0.0.0..2221          *..*                  UDP
```

Netstat Gate/-g report

Displays the IPv4 routing information that this stack uses when it determines what addresses it can communicate with and over which links and first hops the communication takes place. The routes in the stack routing table can be static routes (those defined in the TCP/IP profile), routes learned from routing daemons, and routes learned by other ICMP information, such as redirects. If there is not a route that covers the destination IP address and if there is no DEFAULT route defined, then this stack cannot communicate with that destination. Multiple routes to the same destination, referred to as multipath routes, are also displayed. If multipath is not enabled on the IPCONFIG statement, then the first active route to the destination is always used.

TSO syntax:

```
►►──NETSTAT Gate──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────►◄
```

Modifier:

```
►►──DETAIL──────────────────────────────────────────────────────────►◄
```

DETAIL
: Displays the general IPv4 routing information, the metric or cost of use for the route, and the MVS specific configured parameters for each route.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
               ┌────────────────────┐
               ▼                    │
►►──IPAddr──┬─ipaddr──────────────┬─┴────────────────────────────────►◄
            └─ipaddr/subnetmask─┘
```

z/OS UNIX syntax:

```
►►──netstat -g──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────►◄
```

Modifier:

```
►►──DETAIL──────────────────────────────────────────────────────────►◄
```

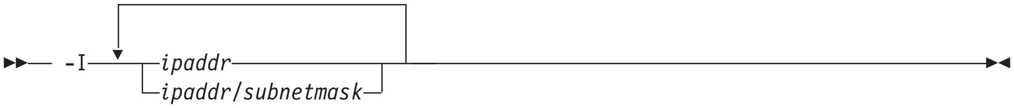
DETAIL
: Displays the general IPv4 routing information, plus the metric or cost of use for the route, and the MVS specific configured parameters for each route.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Filter:



Filter description:

IPAddr/-I *ipaddripaddr/subnetmask*
: Filter the report output using the specified IP address *ipaddr* or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length.

ipaddr Filter the output of the Gate/**-g** report using the specified IP address *ipaddr*. The default subnet mask is 255.255.255.255.

ipaddr/subnetmask
: Filter the output of the Gate/**-g** report using the specified IP address and subnet mask *ipaddr/subnetmask*.

The IPAddr/**-I** filter value can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/**-I** filter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -g -I '10.*.0.0'** or **netstat -g -I "10.*.0.0"**.

Note:

1. The filter value *ipaddr* is the destination IP address; it is not the destination network address.
2. When filtering Gate/**-g** responses on a specified IP address, the DEFAULT and DEFAULTNET routes are not displayed.

Command syntax examples:

From TSO environment:

```
NETSTAT GATE
   Display the routing information the default stack will use when it determines what
   addresses it can communicate with and over which links/interfaces and first hops the
   communication will take place.
NETSTAT GATE TCP TCPCS6
   Display the routing information the TCPCS6 stack will use when it determines what
   addresses it can communicate with and over which links/interfaces and first hops the
   communication will take place.
NETSTAT GATE TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display the routing information in the TCPCS8 stack whose destination address match
   the specified filter IP address values.
```

From UNIX shell environment:

```
   netstat -g
   netstat -g -p tcpcs6
   netstat -g -p tcpcs8 -I 9.43.1.1 9.43.2.2
```

Report examples:

The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT GATE
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          14:50:17
Known gateways:
NetAddress      FirstHop        Link     Pkt Sz Subnet Mask     Subnet Value
----------      --------        ----     ------ -----------     ------------
Default         9.67.113.1      TR1      576    <none>
9.67.1.9        <direct>        OSA00LIN 0      HOST
9.0.0.0         <direct>        TR1      576    0.255.255.128   0.67.113.0
9.67.113.43     <direct>        TR1      17914  HOST
127.0.0.1       <direct         LOOPBACK 65535  HOST
198.11.25.104   198.11.22.109   LMCH2IT2 26624  HOST
201.2.10.31     <direct>        VIPLC902 65535  HOST
```

```
NETSTAT GATE DETAIL
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          14:50:17
Known gateways:
NetAddress      FirstHop        Link     Pkt Sz Subnet Mask     Subnet Value
----------      --------        ----     ------ -----------     ------------
Default         9.67.113.1      TR1      576    <none>
  Metric:  00000000  Flags: UHS
  MVS Specific Configured parameters:
    MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
    RoundTripGain:      0.125     VarianceGain:      0.250
    VarianceMultiplier: 2.000
.....
```

Report field descriptions:

NetAddress

The address of the network. This is the network portion of the destination address of the route. If the route is for a Class A address, then this field contains only the first portion of the address because the class A net mask is 255.0.0.0. If the route is for a Class B address, then this field contains the first half of the address because the class B net mask is 255.255.0.0. If the route is for a Class C route, then this field contains the first 3 parts of the address because the class C net mask is 255.255.255.0.

FirstHop
The first hop address used to send packets to the destination. If <direct>, then the destination is directly reachable without needing to go through a gateway.

Link The link or interface name for the route.

Restriction: Only the first eight characters of the link or interface name are displayed by this command. Issue the NETSTAT ROUTE command to display more than eight characters of the link or interface name.

Pkt Sz This value is the largest packet size that can be sent using this route. If the packet is larger than this size, the packet will have to be fragmented if fragmentation is permitted. If fragmentation is not permitted, the packet would be dropped and an ICMP error would be returned to the originator of the packet.

Subnet Mask
The subnet mask of the network. This is the subnet-only mask for the route. It does not include the class net mask. For example, if the route was for 9.67.114.0 with a net mask of 255.255.255.0 the subnet mask would be 0.255.255.0 because you would not include the class A net mask. Valid values for this field include:

Dotted Decimal Value
This is the subnet-only portion of the net mask. If you take the route's net mask and remove the class mask from it, you are left with the subnet-only portion of the displayed net mask. If you combine the class mask with this field you get the complete net mask for this route entry.

<none>
If this field contains <none>, then this is a network route and the net mask is the class mask for the route destination.

REDIRECT_HOST
This means that this route is for a HOST entry and was learned by an ICMP redirect. The subnet mask would be 255.255.255.255.

HOST This means that this route is for a HOST entry. The subnet mask would be 255.255.255.255.

Subnet Value
The subnet value of the network. This is the subnet portion of the route's destination address. It does not contain the network portion that was displayed in the Address of the network. Valid values for this field include:

Dotted Decimal Value
This is the subnet/host portion of the route's destination address. If you combine this field with the value in Address of the network, you get the complete route destination address.

blank If this field is blank, then this is a network route and the subnet/host portion of the route destination address is zero.

Metric This value displays the metric of the route. For static routes, all direct routes will have a metric of 0 and indirect routes will have a metric of 1. If the routes were learned from a routing daemon, then the metric displayed would be the metric set by the routing daemon. Once the routes are in the stack routing table, the metric field is not used. The routing daemons use metrics to compare routes and inform the stack only of the route or routes that have the best metric.

Flags Identifies the state of the route and can have the following values:

U The route is up.

H The route is to a host rather than to a network.

G The route uses a gateway.

The following flags are mutually exclusive:

C The route was created by a connection (not using a definition or a routing protocol). Routes to subnets or point-to-point destinations using interfaces over which OMPROUTE is active but has not yet established a routing protocol are considered connection routes.

D The route was created dynamically by ICMP processing.

O The route was created by OSPF (includes OSPF external routes).

R The route was created by RIP.

S The route is a static route not replaceable by a routing daemon.

Z The route is a static route replaceable by dynamic routes learned by OMPROUTE.

Maximum retransmit time
The TCP retransmission interval for this route. If this parameter was not specified on the GATEWAY statement, the default value of 120 seconds is displayed. This parameter does not affect initial connection retransmission.

Minimum retransmit time
The minimum retransmit interval for this route. If this parameter was not specified on the GATEWAY statement, the default value of 0.5 (500 milliseconds) seconds is displayed.

Round trip gain
This value is the percentage of the latest round trip time (RTT) to be applied to the smoothed RTT average. The higher this value, the more influence the latest packet RTT has on the average. If this parameter was not specified on the GATEWAY statement, the default value of 0.125 is displayed. This parameter does not affect initial connection retransmission.

Variance gain
This value is the percentage of the latest RTT variance from the RTT average to be applied to the RTT variance average. The higher this value, the more influence the latest packet's RTT has on the variance average. If this parameter was not specified on the GATEWAY statement, the default value of 0.25 is displayed. This parameter does not affect initial connection retransmission.

Variance multiplier
This value is multiplied against the RTT variance in calculating the retransmission interval. The higher this value, the more effect variation in RTT has on calculating the retransmission interval. If this parameter was not specified on the GATEWAY statement, the default value of 2 is displayed. This parameter does not affect initial connection retransmission.

Netstat HElp/-? report

Displays help information for Netstat parameters.

TSO syntax:

```
►►──NETSTAT──┬─HElp─┬────────────────────────────────────────────────►◄
             └─?────┘
```

z/OS UNIX syntax:

```
►►──netstat -?──────────────────────────────────────────────────────────►◄
```

Command syntax examples:

From TSO environment:

```
NETSTAT HELP or NETSTAT ?
```

From UNIX shell environment:

```
   netstat -?
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT HELP or
NETSTAT ?
Usage: NETSTAT <Report option | Command> <Target> <Output> <(Filter>
Report option:
ALL       - Display detailed information about TCP connection
            and UDP sockets
ALLConn   - Display information for all TCP connections and
            UDP sockets, including some recently closed ones
ARp       - Query ARP table or entry information (IPv4 only)
BYTEinfo  - Display the byte-count information for each
            active TCP connection and UDP socket
CACHinfo  - Display information about TCP connections
            utilizing the Cache Accelerator
CLients   - Display information about local users of TCP/IP
            services (jobnames)
CONFIG    - Display the TCP/IP configuration information
COnn      - Display information about each active TCP
            connection and UDP socket (Default option)
DEFADDRT  - Display the policy table for IPv6 default address selection
DEvlinks  - Display information about devices and defined
            interfaces or links
Gate      - Display information about the stack routing table
            for IPv4 destinations
HElp or ? - Display Netstat parameters list
HOme      - Display information about each home IP address
            and its associated link or interface name
IDS       - Display information about Intrusion Detection
            Services
ND        - Display the IPv6 Neighbor cache entries
PORTList  - Display port reservation list
RESCache  - Display resolver cache information
ROUTe     - Display stack routing information
SLAP      - Display QoS policy statistics
SOCKets   - Display information about each client using a
            socket application programming interface
SRCIP     - Displays information for all job-specific source
            VIPA IP address associations
STATS     - Display TCP/IP statistics
TTLS      - Display Application Transparent Transport Layer
            Security (AT-TLS) information
TELnet    - Display TN3270 Telnet server connections
Up        - Date and time tcpip was last started
VCRT      - Display the dynamic VIPA Connection Routing Table
VDPT      - Display the dynamic VIPA Destination Port Table
VIPADCFG  - Display the dynamic VIPA configuration information
VIPADyn   - Display the current dynamic VIPA and VIPAROUTE
            information
Target:
TCP       - Display detailed information about the specified
            TCPIP address space
Output:
FORMat    - Display Netstat report in a given format
REPort    - Netstat information written to dataset name
            tsoprefix.NETSTAT.option or specified with DSN/HLQ
STACk     - Netstat information written to a TSO data stack
```

```
Filter:
APPLD     - Filter the output of ALL,ALLCONN,and CONN reports
            using the specified application data
APPLName  - Filter the output of the TELNET report using the
            specified VTAM application name
CLIent    - Filter the output of ALL, ALLCONN, BYTEINFO, CLIENT,
            CONN, SOCKETS, and TELNET reports using the specified
            client name
CONNType  - Filter the output of ALLCONN and CONN reports using
            the specified connection type
DNSAddr   - Filter the output of RESCACHE using the specified
            DNS IP address.
HOSTNAME  - Filter the output of ALL, ALLCONN, BYTEINFO, CONN, RESCACHE,
            SOCKETS, TELNET and VCRT reports using the specified
            host name
INTFNAME  - Filter the output of DEVLINKS and HOME reports using the
            specified name
IPAddr    - Filter the output of ALL, ALLCONN, BYTEINFO, CONN, GATE,
            ND, RESCACHE, ROUTE, SOCKETS, TELNET, VCRT, VDPT, and VIPADCFG
            reports using the specified IP address
IPPort    - Filter output of the ALL, ALLCONN, CONN, SOCKETS,
            TELNET, VCRT, and VDPT reports using the specified IP
            address and port number
LUName    - Filter the output of the TELNET report using the
            specified LU name
NOTN3270  - Filter the output of ALL, ALLCONN, BYTEINFO, CONN,
            CLIENTS, and SOCKETS reports excluding TN3270 server
            connections
POLicyn   - Filter the output of the SLAP report using the specified
            policy name
POrt      - Filter the output of ALL, ALLCONN, CONN, PORTLIST, SOCKETS,
            TELNET, VCRT, and VDPT reports using the specified port
SMCID   -   Filter the output of ALL, ALLConn, CONN, and DEVLINKS reports
            using the specified SMC-R link or link group identifier
Command:
DRop      - Terminates the socket end-point that is identified by
            the specified connection number
```

```
netstat -?
Usage: netstat|onetstat <Report Option | Command> <Target> <Output> <Filter>
Report option:
-A  - Display detailed information about TCP connection and UDP
      sockets
-a  - Display information for all TCP connections and UDP sockets,
      including some recently closed ones
-b  - Display the byte-count information for each active TCP
      connection and UDP socket
-C  - Display information about TCP connections utilizing the
      Cache Accelerator
-c  - Display information about each active TCP connection and UDP
      socket (Default option)
-d  - Display information about devices and defined interface or
      links
-e  - Display information about local users of TCP/IP services
      (jobname)
-F  - Display the dynamic VIPA configuration information
-f  - Display the TCP/IP configuration information
-g  - Display information about the stack routing table for IPv4
      destinations
-h  - Display information about each home IP address and its
      associated link or interface name
-J  - Displays information for all job-specific source VIPA IP
      address associations
-j  - Display QoS policy statistics
-k  - Display information about Intrusion Detection Services
-l  - Display the policy table for IPv6 default address selection
-n  - Display the IPv6 Neighbor cache entries
-O  - Display the dynamic VIPA Destination Port Table
-o  - Display port reservation list
-q  - Display resolver cache information
-R  - Query ARP table or entry information (IPv4 only)
-r  - Display stack routing information
-S  - Display TCP/IP statistics
-s  - Display information about each client using socket
      application programming interface
-t  - Display TN3270 Telnet server connections
-u  - Date and time tcpip was last started
-V  - Display the dynamic VIPA Connection Routing Table
-v  - Display the current dynamic VIPA and VIPAROUTE information
-x  - Display Application Transparent Transport Layer
      Security (AT-TLS) information
-?  - Display Netstat parameters list
Target:
-p  - Display detailed information about the specified
      TCPIP address space
Output:
-M  - Display Netstat report in a given format
```

```
Filter:
-B    Filter output of the -A, -a, -c, -s, -t, -O, and -V reports
      using the specified IP address and port number
-E    Filter the output of -A, -a, -b, -c, -e, -s, and -t reports
      using the specified client name
-G    Filter the output of -A, -a, and -c reports using the specified
      application data
-H    Filter the output of -A, -a, -b, -c, -q, -s, -t, and -V reports
      using the specified host name
-I    Filter the output of -A, -a, -b, -c, -F, -g, -n, -q, -r, -s, -t, -O,
      and -V reports using the specified IP address
-K    Filter the output of -d and -h reports using the specified name
-L    Filter the output of -t report using the specified LU name
-N    Filter the output of -t report using the specified application
      name
-P    Filter the output of -A, -a, -c, -s, -t, -O, -o and -V reports
      using the specified port
-Q    Filter the output of -q report using the specified DNS IP address.
-T    Filter the output of -A, -a, -b, -c, -e, and -s reports
      excluding TN3270 server connections
-U    Filter the output of -A, -a, -c, and -d reports using the specified
      SMC-R link or link group identifier
-X    Filter the output of -a, and -c reports using the specified connection
      type
-Y    Filter the output of -j report using the specified policy name
Command:
-D  - Terminates the socket end-point that is identified by the
      specified connection number
```

Netstat HOme/-h report

Displays information about each home IP address and its associated link or interface name.

For more information about the home list, see the z/OS Communications Server: IP Configuration Reference.

TSO syntax:

```
►►──NETSTAT HOme──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────────────────►◄
```

Target:

Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
►►──INTFName──intfname──────────────────────────────────────────────────────────►◄
```

z/OS UNIX syntax:

```
►►──netstat -h──┤ Target ├──┤ Output ├──┤ Filter ├─────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
►►— -K—intfname————————————————————————————————————————————————►◄
```

Filter description:

INTFName/-K *intfname*

Filter the output of the HOme/**-h** report using the specified interface name *intfname*. You can enter one filter value at a time and the specified value can be up to 16 characters long.

The INTFName filter value *intfname* can be one of the following names:

- The link name of a network interface that was configured on a LINK profile statement (this option selects one interface).
- The interface name of a network interface that was configured on an INTERFACE profile statement (this option selects one interface).
- The port name of an OSA-Express feature in QDIO mode, where the port name value is the name that is specified on the PORTNAME keyword in the TRLE (this option selects all interfaces that are associated with the OSA-Express port).
- The name of a HiperSockets TRLE. This option selects all interfaces that are associated with the HiperSockets TRLE.

Restriction: The INTFName/**-K** filter value does not support wildcard characters.

Command syntax examples:

From TSO environment:

```
NETSTAT HOME
   Display the home list information for the default stack. If the stack is IPv6-enabled,
   then both IPv4 and IPv6 home list information are displayed.
NETSTAT HOME TCP TCPCS6
   Display the home list information for the TCPCS6 stack. If the TCPCS6 stack is
   IPv6-enabled, then both IPv4 and IPv6 home list information are displayed.
NETSTAT HOME TCP TCPCS8 (INTFNAME OSAQDIOLINK
   Display the home list information for the OSAQDIOLINK in the TCPCS8 TCP/IP adress space.
```

From UNIX shell environment:

```
   netstat -h
   netstat -h -p tcpcs6
   netstat -h -p tcpcs8 -K
```

Report examples:

The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT HOME
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:41:00
Home address list:
Address          Link             Flg
-------          ----             ---
9.67.115.5       OSAQDIOLINK      P
9.67.113.11      TR1
201.2.10.31      VIPLC9020A1F     I
127.0.0.1        LOOPBACK

Address          Interface        Flg
-------          ---------        ---
9.2.2.2          VIRTUAL2
9.67.125.5       OSAQDIOINTF
9.1.1.1          HIPERSOCK1
```

```
NETSTAT HOME
EZZ2350I MVS TCP/IP NETSTAT CS V2R1        TCPIP Name: TCPCS           19:43:50
EZZ2700I Home address list:
EZZ2701I Address          Link             Flg
EZZ2702I -------          ----             ---
EZZ2703I 10.220.0.1       EZASAMEMVS
EZZ2703I 127.0.0.1        LOOPBACK

EZZ2704I Address          Interface        Flg
EZZ2704I -------          ---------        ---
EZZ2703I 0.0.0.0          LOSAQDIO2        PI
```

IPv6 enabled or request for LONG format:

```
NETSTAT HOME
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:23:53
Home address list:
LinkName:   OSAQDIOLINK
  Address:  9.67.115.5
    Flags:  Primary
LinkName:   TR1
  Address:  9.67.113.11
    Flags:
LinkName:   VIPLC9020A1F
  Address:  201.2.10.31
    Flags:  Internal
LinkName:   LOOPBACK
  Address:  127.0.0.1
    Flags:
IntfName:   VIRTUAL2
  Address:  9.2.2.2
    Flags:
IntfName:   OSAQDIOINTF
  Address:  9.67.125.5
    Flags:
IntfName:   HIPERSOCK1
  Address:  9.1.1.1
    Flags:
IntfName:   VIPAV6
  Address:  2001:0db8::a:9:67:115:5
    Type:   Global
    Flags:
  Address:  50c9:c2d4:0:a:9:67:115:5
    Type:   Global
    Flags:  Deprecated
IntfName:   OSAQDIO46
  Address:  2001:0db8::9:67:125:5
    Type:   Global
    Flags:
  Address:  fe80::6:2900:1dc:217c
    Type:   Link_Local
    Flags:  Autoconfigured
IntfName:   OSAQDIO48
  Address:  fe80::6:2900:6dc:217c
    Type:   Link_Local
    Flags:  Autoconfigured
  Address:  50c9:c2d4::6:2900:6dc:217c
    Type:   Global
    Flags:  Autoconfigured
  Address:  50c9:c2d4::a8ed:838d:c853:3832
    Type:   Global
    Flags:  Autoconfigured,Temporary
    ValidLifetimeExp: 08/28/2011 15:35
  Address:  50c9:c2d4::5757:3772:9494:9944
    Type:   Global
    Flags:  Autoconfigured,Temporary,Deprecated
    ValidLifetimeExp: 08/21/2011 16:36

IntfName:   LOOPBACK6
  Address:  ::1
    Type:   Loopback
    Flags:

Unavailable IPv6 Home addresses:
IntfName:   OSAQDIO26
  Address:  2001:0db8::9:67:115:66
    Type:   Global
    Reason: Duplicate address detection pending start of interface
  Address:  2001:0db8::/64
    Type:   Global
    Reason: Interface ID not yet known
```

```
NETSTAT HOME
MVS TCP/IP NETSTAT CS V2R1        TCPIP Name: TCPCS          19:47:19
 Home address list:
 LinkName:   EZASAMEMVS
   Address:  10.220.0.1
     Flags:
 LinkName:   LOOPBACK
   Address:  127.0.0.1
     Flags:
 IntfName:   LOSAQDIO2
   Address:  0.0.0.0
     Flags:  Primary Internal
```

Report field descriptions:

For a SHORT format report:

Address
IPv4 address for this home entry.

Link Link name for this home entry.

Flg Flags, which include the following values:

P Primary interface.

I One of the following flags:

- An internally generated dynamic VIPA that is not advertised to routing daemons or an interface that is defined with the TEMPIP keyword. This flag is displayed for the following VIPAs or interfaces:
  - – Dynamic VIPAs that are created on target stacks for the sysplex distributor
  - – Dynamic VIPAs on stacks that are the endpoint for connections where the dynamic VIPA has moved to another stack
  - – IPAQENET interfaces that are defined with the TEMPIP keyword
- An interface that is defined with the TEMPIP keyword. This flag is displayed for IPAQENET interfaces that are defined with the TEMPIP keyword.

For a LONG format report:
For an IPv4 home list entry:

Address
IPv4 address for this home entry.

LinkName
Link name for this home entry.

Flags

Primary
Primary interface.

Internal
One of the following flags:

- An internally generated dynamic VIPA that is not advertised to routing daemons or an interface that is defined with the TEMPIP keyword. This flag is displayed for the following dynamic VIPAs or interfaces:
  - Dynamic VIPAs that are created on target stacks for the sysplex distributor
  - Dynamic VIPAs on stacks that are the endpoint for connections where the dynamic VIPA has moved to another stack
  - IPAQENET interfaces that are defined with the TEMPIP keyword
- An interface that is defined with the TEMPIP keyword. This flag is displayed for IPAQENET interfaces that are defined with the TEMPIP keyword.

For an IPv6 home list entry:

IntfName
Interface name for this home entry.

Address
IPv6 address for this home entry.

Type Address type that can be Global, Loopback, or Link_Local.

Flags

Autoconfigured
The IP address was built from prefix information supplied by the router.

Deprecated
The preferred lifetime of the autoconfigured address has expired.

Internal
An internally generated VIPA that is not advertised to routing daemons.

Temporary
A temporary IP address that was built from prefix information that was supplied by the router, and a randomly generated interface ID.

ValidLifetimeExp
The time at which the IPv6 temporary autoconfigured address will be deleted. This valid lifetime can be extended by router-supplied information. This field is displayed only for a temporary IP address.

For an IPv6-enabled stack, the unavailable IPv6 home addresses are also displayed, which contain the following information for each entry in the list:

IntfName
Interface name for this home entry.

Address
IPv6 address for this home entry.

Type Address type including Global, Loopback, or Link_Local.

Reason
Reason the IP address is unavailable:

Duplicate address detection in progress
: Duplicate address detection is in progress to determine if another node is currently using the IP address. The IP address is made available if it is determined to be unique on the local link.

Duplicate address detected
: Duplicate address detection was previously done for this IP address and the IP address was in use elsewhere.

Duplicate address detection pending start of interface
: Duplicate address detection has been requested for the interface but the interface has not been started. The interface must be started before duplicate address detection can be done and this IP address made available.

Duplicate address detection prevented by IPSec
: Duplicate address detection has been requested for the interface, but the outbound Neighbor Solicitation packet has been denied by IPSec policy.

Interface ID not yet known
: This reason applies to interfaces for which duplicate address detection will not be performed (for example, where a value of 0 was configured for the DUPADDRDET parameter) and where an IP address prefix was configured. An interface ID is required to append to the prefix to create the full IP address. The interface ID is not available until the interface is successfully started.

Netstat IDS/-k report

Displays information about intrusion detection services.

TSO syntax:

```
►►──NETSTAT IDS──┤ Modifier ├──┤ Target ├──┤ Output ├─────────────────────►◄
```

Modifier:

```
►►──┬────────────────────┬──────────────────────────────────────────────►◄
    ├─SUMmary────────────┤
    └─PROTOcol──protocol─┘
```

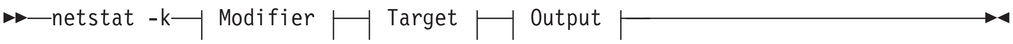
SUMmary
: Displays summary information about intrusion detection services.

PROTOcol *protocol*
: Displays information about intrusion detection services for the specified protocol. The valid protocols are TCP and UDP.

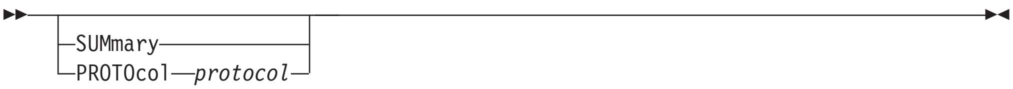
Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

z/OS UNIX syntax:



Modifier:



SUMmary
Displays summary information about intrusion detection services.

PROTOcol *protocol*
Displays information about intrusion detection services for the specified protocol. The valid protocols are TCP and UDP.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT IDS
NETSTAT IDS SUMMARY
NETSTAT IDS PROTOCOL TCP
NETSTAT IDS PROTOCOL UDP
```

From UNIX shell environment:

```
   netstat -k
   netstat -k SUMMARY
   netstat -k PROTOCOL TCP
   netstat -k PROTOCOL UDP
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX Netstat command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled example (SHORT format):

```
NETSTAT IDS
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:51:44
Intrusion Detection Services Summary:
Scan Detection:
  GlobRuleName: ScanGlobal-rule
  IcmpRuleName: ScanEventIcmp-rule
  TotDetected:  0           DetCurrPlc: 0
  DetCurrInt:   0           Interval:   60
  SrcIPsTrkd:   0           StrgLev:    00000
Attack Detection:
  Malformed Packets
    PlcRuleName: AttackMalformed-rule
    TotDetected: 11          DetCurrPlc: 8
    DetCurrInt:  0           Interval:   0
  OutBound RAW Restrictions
    PlcRuleName: AttackOutboundRaw-rule
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  Restricted Protocols
    PlcRuleName: AttackIPprot-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  Restricted IP Options
    PlcRuleName: AttackIPopt-rule
    TotDetected: 64          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  ICMP Redirect Restrictions
    PlcRuleName: AttackICMPRedirect-rule
    TotDetected: 10          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  IP Fragment Restrictions
    PlcRuleName: AttackIpFragment-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  UDP Perpetual Echo
    PlcRuleName: AttackPerpEcho-rule
    TotDetected: 32          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  Floods
    PlcRuleName: AttackFlood-rule
    TotDetected: 3           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   5
  Data Hiding
    PlcRuleName: AttackDataHiding-rule
    TotDetected: 8           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  TCP Queue Size
    PlcRuleName: AttackTCPQueSz-rule
    TotDetected: 27          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  Global TCP Stall
    PlcRuleName: AttackTCPStall-rule
    TotDetected: 1           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  EE LDLC Check
    PlcRuleName: EE_Attack-LDLC
    TotDetected: 3           DetCurrPlc: 3
    DetCurrInt: 0            Interval:   60
  EE Malformed Packet
    PlcRuleName: EE_Attack-Malformed
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  EE Port Check
    PlcRuleName: EE_Attack-Port
    TotDetected: 2           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   60
  EE XID Flood
    PlcRuleName: EE_Attack-XID
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   60
Traffic Regulation:
  TCP
    ConnRejected: 3           PlcActive: Y
  UDP
    PckDiscarded: 0           PlcActive: Y
Active Global Conditions:
  ServersInConnFlood: 5
  TCPStalledConns: 345            TCPStalledConnsPct: 14
Active Interface Floods:
  IntfName: ETH1
    DiscardCnt: 1828        DiscardRate: 57   Duration: 68
Intrusion Detection Services TCP Port List:
TcpListeningSocket: 0.0.0.0..23
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: ids-rule1
  TrPortInst: Y  TrCorr: 0           MxApp: 0           MxHst: 3
  SynFlood:   N  ConnFlood: N
Intrusion Detection Services UDP Port List:
UdpDestSocket: 9.39.69.147..909
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: *NONE*
  TrCorr: 0           Discarded: 0
```

```
NETSTAT IDS SUMMARY
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:51:44
Intrusion Detection Services Summary:
Scan Detection:
  GlobRuleName: ScanGlobal-rule
  IcmpRuleName: ScanEventIcmp-rule
  TotDetected:  0           DetCurrPlc: 0
  DetCurrInt:   0           Interval:   60
  SrcIPsTrkd:   0           StrgLev:    00000
Attack Detection:
  Malformed Packets
    PlcRuleName: AttackMalformed-rule
    TotDetected: 11          DetCurrPlc: 8
    DetCurrInt:  0           Interval:   0
  OutBound RAW Restrictions
    PlcRuleName: AttackOutboundRaw-rule
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  Restricted Protocols
    PlcRuleName: AttackIPprot-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  Restricted IP Options
    PlcRuleName: AttackIPopt-rule
    TotDetected: 64          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  ICMP Redirect Restrictions
    PlcRuleName: AttackICMPRedirect-rule
    TotDetected: 10          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  IP Fragment Restrictions
    PlcRuleName: AttackIpFragment-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  UDP Perpetual Echo
    PlcRuleName: AttackPerpEcho-rule
    TotDetected: 32          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  Floods
    PlcRuleName: AttackFlood-rule
    TotDetected: 3           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   5
  Data Hiding
    PlcRuleName: AttackDataHiding-rule
    TotDetected: 8          DetCurrPlc: 0
    DetCurrInt:  0          Interval:   0
  TCP Queue Size
    PlcRuleName: AttackTCPQueSz-rule
    TotDetected: 27         DetCurrPlc: 4
    DetCurrInt:  0          Interval:   0
  Global TCP Stall
    PlcRuleName: AttackTCPStall-rule
    TotDetected: 1          DetCurrPlc: 0
    DetCurrInt:  0          Interval:   0
  EE LDLC Check
    PlcRuleName: EE_Attack-LDLC
    TotDetected: 3          DetCurrPlc: 3
    DetCurrInt:  0          Interval:   60
  EE Malformed Packet
    PlcRuleName: EE_Attack-Malformed
    TotDetected: 0          DetCurrPlc: 0
    DetCurrInt:  0          Interval:   0
  EE Port Check
    PlcRuleName: EE_Attack-Port
    TotDetected: 2          DetCurrPlc: 2
    DetCurrInt:  0          Interval:   60
  EE XID Flood
    PlcRuleName: EE_Attack-XID
    TotDetected: 0          DetCurrPlc: 0
    DetCurrInt:  0          Interval:   60
Traffic Regulation:
  TCP
    ConnRejected: 3          PlcActive: Y
  UDP
    PckDiscarded: 0          PlcActive: Y
Active Global Conditions:
  ServersInConnFlood: 5
  TCPStalledConns: 345          TCPStalledConnsPct: 14
Active Interface Floods:
  IntfName: ETH1
    DiscardCnt: 1828       DiscardRate: 57   Duration: 68
```

```
NETSTAT IDS PROTOCOL TCP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          11:51:44
Intrusion Detection Services TCP Port List:
TcpListeningSocket: 0.0.0.0..23
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: ids-rule1
  TrPortInst: Y  TrCorr: 0          MxApp: 0          MxHst: 3
  SynFlood:   N  ConnFlood: N
```

```
NETSTAT IDS PROTOCOL UDP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          11:51:44
Intrusion Detection Services UDP Port List:
UdpDestSocket: 9.39.69.147..909
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: *NONE*
  TrCorr: 0          Discarded: 0
```

IPv6 enabled or request for LONG format:

```
NETSTAT IDS
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:51:44
Intrusion Detection Services Summary:
Scan Detection:
  GlobRuleName:   ScanGlobal-rule
  IcmpRuleName:   ScanEventIcmp-rule
  Icmpv6RuleName: ScanEventIcmpv6-rule
  TotDetected:  0           DetCurrPlc: 0
  DetCurrInt:   0           Interval:   60
  SrcIPsTrkd:   0           StrgLev:    00000
Attack Detection:
  Malformed Packets
    PlcRuleName: AttackMalformed-rule
    TotDetected: 11          DetCurrPlc: 8
    DetCurrInt:  0           Interval:   0
  OutBound IPv4 RAW Restrictions
    PlcRuleName: AttackOutboundRaw-rule
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  Restricted IPv4 Protocols
    PlcRuleName: AttackIPprot-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  Restricted IPv4 Options
    PlcRuleName: AttackIPopt-rule
    TotDetected: 64          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  ICMP Redirect Restrictions
    PlcRuleName: AttackICMPRedirect-rule
    TotDetected: 10          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  IP Fragment Restrictions
    PlcRuleName: AttackIpFragment-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  UDP Perpetual Echo
    PlcRuleName: AttackPerpEcho-rule
    TotDetected: 32          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  Floods
    PlcRuleName: AttackFlood-rule
    TotDetected: 3           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   5
  Data Hiding
    PlcRuleName: AttackDataHiding-rule
    TotDetected: 8           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  TCP Queue Size
    PlcRuleName: AttackTCPQueSz-rule
    TotDetected: 27          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  Global TCP Stall
    PlcRuleName: AttackTCPStall-rule
    TotDetected: 1           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  EE LDLC Check
    PlcRuleName: EEAttack-LDLC
    TotDetected: 3           DetCurrPlc: 3
    DetCurrInt:  0           Interval:   60
  EE Malformed Packet
    PlcRuleName: EEAttack-Malformed
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   60
  EE Port Check
    PlcRuleName: EE_Attack-Port
    TotDetected: 2           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   60
  EE XID Flood
    PlcRuleName: EE_Attack-XID
    TptDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   60
  OutBound IPv6 RAW Restrictions
    PlcRuleName: AttackOutboundv6Raw-rule
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  Restricted IPv6 Next Headers
    PlcRuleName: AttackNextHdr-rule
    TotDetected: 30          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  Restricted IPv6 Destination Options
    PlcRuleName: AttackDestOpts-rule
    TotDetected: 15          DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  Restricted IPv6 Hop-by-Hop Options
    PlcRuleName: AttackHopOpts-rule
    TotDetected: 3           DetCurrPlc: 1
    DetCurrInt:  0           Interval:   0
Traffic Regulation:
  TCP
    ConnRejected: 3           PlcActive: Y
  UDP
    PckDiscarded: 0           PlcActive: Y
Active Global Conditions:
  ServersInConnFlood: 5
  TCPStalledConns: 345            TCPStalledConnsPct: 14
Active Interface Floods:
  IntfName: ETH1
        DiscardCnt: 1828        DiscardRate: 57   Duration: 68
Intrusion Detection Services TCP Port List:
TcpListeningSocket: 0.0.0.0..23
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: ids-rule1
  TrPortInst: Y  TrCorr: 0           MxApp: 0           MxHst: 3
  SynFlood:   N  ConnFlood: N
TcpListeningSocket: 2001:db8::9:67:115:66..21
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: ids-rule1
  TrPortInst: Y  TrCorr: 0           MxApp: 1           MxHst: 2
  SynFlood:   N  ConnFlood: N
Intrusion Detection Services UDP Port List:
UdpDestSocket: 9.39.69.147..909
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: *NONE*
  TrCorr: 0           Discarded: 0
UdpDestSocket: 2001:db8::9:67:115:78..911
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: *NONE*
  TrCorr: 0           Discarded: 0
```

```
NETSTAT IDS SUMMARY
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:51:44
Intrusion Detection Services Summary:
Scan Detection:
  GlobRuleName:   ScanGlobal-rule
  IcmpRuleName:   ScanEventIcmp-rule
  Icmpv6RuleName: ScanEventIcmpv6-rule
  TotDetected:  0           DetCurrPlc: 0
  DetCurrInt:   0           Interval:   60
  SrcIPsTrkd:   0           StrgLev:    00000
Attack Detection:
  Malformed Packets
    PlcRuleName: AttackMalformed-rule
    TotDetected: 11          DetCurrPlc: 8
    DetCurrInt:  0           Interval:   0
  OutBound IPv4 RAW Restrictions
    PlcRuleName: AttackOutboundRaw-rule
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  Restricted IPv4 Protocols
    PlcRuleName: AttackIPprot-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  Restricted IPv4 Options
    PlcRuleName: AttackIPopt-rule
    TotDetected: 64          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  ICMP Redirect Restrictions
    PlcRuleName: AttackICMPRedirect-rule
    TotDetected: 10          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  IP Fragment Restrictions
    PlcRuleName: AttackIpFragment-rule
    TotDetected: 4           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  UDP Perpetual Echo
    PlcRuleName: AttackPerpEcho-rule
    TotDetected: 32          DetCurrPlc: 10
    DetCurrInt:  0           Interval:   0
  Floods
    PlcRuleName: AttackFlood-rule
    TotDetected: 3           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   5
  Data Hiding
    PlcRuleName: AttackDataHiding-rule
    TotDetected: 8           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  TCP Queue Size
    PlcRuleName: AttackTCPQueSz-rule
    TotDetected: 27          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  Global TCP Stall
    PlcRuleName: AttackTCPStall-rule
    TotDetected: 1           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  EE LDLC Check
    PLCRuleName: EE_Attack-LDLC
    TotDetected: 3           DetCurrPlc: 3
    DetCurrInt:  0           Interval:   1
  EE Malformed Packet
    PlcRuleName: EE_Attack-Malformed
    TotDetected: 0           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  EE Port Check
    PlcRuleName: EE_Attack-Port
    TotDetected: 2           DetCurrPlc: 2
    DetCurrInt:  0           Interval:   60
  EE XID Flood
    PlcRuleName: EE_Attack-XID
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   60
  OutBound IPv6 RAW Restrictions
    PlcRuleName: AttackOutboundv6Raw-rule
    TotDetected: 0           DetCurrPlc: 0
    DetCurrInt:  0           Interval:   0
  Restricted IPv6 Next Headers
    PlcRuleName: AttackNextHdr-rule
    TotDetected: 30          DetCurrPlc: 4
    DetCurrInt:  0           Interval:   0
  Restricted IPv6 Destination Options
    PlcRuleName: AttackDestOpts-rule
    TotDetected: 15          DetCurrPlc: 2
    DetCurrInt:  0           Interval:   0
  Restricted IPv6 Hop-by-Hop Options
    PlcRuleName: AttackHopOpts-rule
    TotDetected: 3           DetCurrPlc: 1
    DetCurrInt:  0           Interval:   0
Traffic Regulation:
  TCP
    ConnRejected: 3           PlcActive: Y
  UDP
    PckDiscarded: 0           PlcActive: Y
Active Global Conditions:
  ServersInConnFlood: 5
  TCPStalledConns: 345        TCPStalledConnsPct: 14
Active Interface Floods:
  IntfName: ETH1
    DiscardCnt: 1828        DiscardRate: 57   Duration: 68
```

```
NETSTAT IDS PROTOCOL TCP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:51:44
Intrusion Detection Services TCP Port List:
TcpListeningSocket: 0.0.0.0..23
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: ids-rule1
  TrPortInst: Y  TrCorr: 0           MxApp: 0           MxHst: 3
  SynFlood:   N  ConnFlood: N
TcpListeningSocket: 2001:db8::9:67:115:66..21
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: ids-rule1
  TrPortInst: Y  TrCorr: 0           MxApp: 1           MxHst: 2
  SynFlood:   N  ConnFlood: N
```

```
NETSTAT IDS PROTOCOL UDP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:51:44
Intrusion Detection Services UDP Port List:
UdpDestSocket: 9.39.69.147..909
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: *NONE*
  TrCorr: 0           Discarded: 0
UdpDestSocket: 2001:db8::9:67:115:78..911
  ScStat: C  ScRuleName: ids-rule7
  TrStat: C  TrRuleName: *NONE*
  TrCorr: 0           Discarded: 0
```

Report field descriptions:

SUMmary

Display summary information about intrusion detection services. The following describes the information displayed by the SUMmary option.

- **For Scan Detection:**

  This section displays the following scan detection information. See Intrusion detection services in z/OS Communications Server: IP Configuration Guide for detailed information about IDS scan support.

  GlobRuleName
  The Global Scan rule name or *NONE* if scan detection is not active.

  IcmpRuleName
  The Scan ICMP rule name or *NONE* if ICMP scan event policy is not active.

  Icmpv6RuleName
  The Scan ICMPv6 rule name or *NONE* if ICMPv6 scan event policy is not active.

  TotDetected
  The number of scans detected since the TCP stack was started.

  DetCurrPlc
  The number of scans detected since the last Scan Global policy change.

  DetCurrInt
  The number of scans detected in the current scan interval.

  Interval
  The length of the internal scan interval used to detect scans. This value is either 30 seconds or 60 seconds depending on the fast scan interval specified in the policy.

  SrcIPsTrkd
  The number of source IP addresses currently being monitored by scan detection.

  StrgLev
  The amount of private storage, in megabytes, that scan detection is using. This value is calculated at each internal interval. If 0 is shown, this indicates that no storage is currently in use for scan detection. 0M indicates that less than 1 MB of storage is in use.

- **For Attack Detection:**

  This section displays the following information for each attack type. See Intrusion Detection Services in z/OS Communications Server: IP Configuration Guide for detailed information about IDS attack support.

  PlcRuleName
  The attack rule name or *NONE* if no policy is active for the attack type.

  TotDetected
  The number of attacks detected since the TCP stack was started.

  DetCurrPlc
  The number of attacks detected since the last policy change.

DetCurrInt
The number of attacks detected in the current statistics interval. If statistics is not specified in the policy, the value of this field is 0.

Interval
The current statistics interval or 0 if statistics is not specified in the policy.

- **For Traffic Regulation:**

  This section displays the following TCP and UDP traffic regulation information. See Intrusion detection services in z/OS Communications Server: IP Configuration Guide for detailed information about IDS traffic regulation support.

  ConnRejected
  The number of TCP connections rejected by Traffic Regulation since the TCP/IP stack was started.

  PckDiscarded
  The number of UDP packets discarded by Traffic Regulation since the TCP/IP stack was started.

  PlcActive

  Y Indicates that TR policy is active for at least one port in the respective protocol.

  N Indicates that Traffic Regulation is not active for any ports in the respective protocol.

- **For Active Global Conditions:**

  Displays the following global state information related to IDS and attack protection.

  ServersInConnFlood
  The number of TCP servers that are currently under a potential connection flood attack. A server is considered under a potential connection flood attack when backlog queue expansion is required to handle the incoming connection requests. When more than 25 servers are under a potential connection flood attack, no server's backlog queue will be allowed to expand. This is an action taken to protect TCP/IP stack resources. There is no IDS configuration associated with this protection.

  TCPStalledConns
  The number of TCP connections whose send data flow is currently stalled. The send data flow is considered stalled if one or more of the following conditions are true:

  – The TCP send window size is less than 256 or is less than the smaller of the largest send window that has been seen for the connection and the default MTU. The TCP send window size is set based on values provided by the TCP peer. The default MTU for IPv4 is 576. The default MTU for IPv6 is 1280.

  – The TCP send queue is full and the data is not being retransmitted.

  TCPStalledConnsPct
  The percentage of active TCP connections whose send data flow is currently stalled. If IDS attack type Global TCP Stall is configured, a global TCP stall condition is detected when the

send data flow of at least 50% of the active TCP connections is stalled and at least 1000 TCP connections are active.

- **For Active Interface Floods:**

  This section is displayed only if there is one or more interface floods in progress. Interface flood discard counts and rates are updated at one-minute intervals.

  Intfname
  The name of the interface that is currently experiencing an interface flood condition.

  DiscardCnt
  The number of inbound packets discarded or not processed since the interface flood was detected.

  DiscardRate
  The percentage of discarded packets detected on the interface since the interface flood was detected.

  Duration
  The number of seconds since the start of the interface flood was detected.

PROTOcol *protocol*
Display information about intrusion detection services for the specified protocol. The valid protocols are TCP and UDP.

The following describes the information displayed by the PROTOcol selected. The information is displayed by destination IP address and port. This information is displayed only for the applications with IDS related information, such as if Traffic Regulation or Scan Detection policy is active for the application. For TCP, the data is also shown if the application is currently experiencing a SYN flood.

TcpListeningSocket
The destination IP address and port.

ScStat ScRuleName currency, can have the following values:

C Indicates ScRuleName shows the most recent Scan event rule for this application.

S Indicates policy has changed and ScRuleName might not yet reflect the change.

ScRuleName
The Scan Event rule associated with this application or *NONE*.

TrStat
TrRuleName currency, can have the following values:

C Indicates TrRuleName shows the most recent Traffic Regulation rule for this application.

S Indicates policy has changed and TrRuleName might not yet reflect the change.

TrRuleName
The Traffic Regulation rule associated with this application or *NONE*.

TrPortInst
If TrRuleName is shown:

Y Indicates that TCP traffic regulation was configured to limit by each socket (also known as limit by port instance). This data applies only to this application.

N Indicates that TCP traffic regulation was not configured to limit by each socket. The MxApp and MxHst information applies to all applications using this port that do not have a separate rule that was configured to limit by each socket.

TrCorr The traffic regulation constrained state correlator. A value of 0 indicates the application is not constrained.

MxHst
The number of connections rejected since the last policy change due to a source IP exceeding the percentage of available connections allowed for a single source IP.

MxApp
The number of connections rejected since the last policy change because the total number of connections was exceeded.

SynFlood

Indicates if the application is currently experiencing a SYN flood. A server is considered under a SYN flood attack when connection requests are being discarded because the backlog queue is full and cannot be expanded any further.

Y Indicates a SYN flood is in progress.

N Indicates a SYN flood is not in progress.

ConnFlood
Indicates if the application is currently experiencing a potential connection flood. A server is considered under a potential connection flood attack when backlog queue expansion is required to handle the incoming connection requests. The point where a potential connection flood attack is detected is based on the initial size of the backlog queue. A small initial backlog queue (for example, 10 entries) is allowed to expand twice before the server is considered under attack, while a server with a large initial backlog queue (for example, 500 entries) can expand once, up to a maximum of 768 entries, before it is considered under attack.

Y Indicates a potential connection flood is in progress.

N Indicates a potential connection flood is not in progress.

UdpDestSocket
The destination IP address and port.

Discarded
The total number of packets discarded since the last policy change because the queue size configured for UDP traffic regulation was exceeded.

Netstat ND/-n report

Displays the IPv6 Neighbor cache entries.

Tip: This report can also be used to display all IPv6 addresses on the HiperSockets internal LAN to which the stack has a route over this interface.

Guideline: For HiperSockets interfaces, the stack requests this data from the appropriate device. If a device does not return this data in a timely fashion, then Netstat will not display data for that interface.

TSO syntax:

```
►►──NETSTAT ND──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
             ┌─────────────────────┐
             ▼                     │
►►──IPAddr────┬─ipaddr───────────┬─┴──────────────────────────────────────────►◄
              └─ipaddr/prefixLen─┘
```

z/OS UNIX syntax:

```
►►──netstat -n──┤ Target ├──┤ Output ├──┤ Filter ├───────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
          ┌─────────────────────┐
          ▼                     │
►►── -I────┬─ipaddr───────────┬─┴─────────────────────────────────────────────►◄
           └─ipaddr/prefixLen─┘
```

Filter description:

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength*
: Filter the report output using the specified IP address *ipaddr* or *ipaddr/prefixlength*. You can enter up to six filter values. Each specified *ipaddr* value must be an IPv6 address that can be up to 45 characters in length.

ipaddr Filter the output of the ND/**-n** report using the specified IP address *ipaddr*. The default *prefixlength* is 128.

ipaddr/prefixlength
Filter the output of the ND/**-n** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv6 address, the prefix length range is 1 – 128.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For the ND/**-n** report, an IPv4 *ipaddr* value is not accepted.
3. For an IPv6-enabled stack, an IPv4-mapped IPv6 address is accepted and is treated as an IPv6 address. If an IPv4-mapped IPv6 address is entered as an IPAddr/**-I** value, there is no matching entry found.

Command syntax examples:

From TSO environment:

```
NETSTAT ND
```

From UNIX shell environment:

```
netstat -n
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
NETSTAT ND
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:33:33
Query Neighbor cache for fe80::202:55ff:fe64:2de7
  Intfname: OSAQDIO46          Intftype: IPAQENET6
  LinklayerAddr: 000255642DE7  State: Stale
  Type: Host                   AdvDfltRt: No
Query Neighbor cache for 2001:0db8::9:67:114:46
  Intfname: OSAQDIO46          Intftype: IPAQENET6
  LinkLayerAddr: 0060CF208827  State: Reachable
  Type: Host                   AdvDfltRt: No
Query Neighbor cache for fe80::206:2aff:fe71:4400
  IntfName: OSAQDIO46          IntfType: IPAQENET6
  LinkLayerAddr: 00062A714400  State: Reachable
  Type: Route                  AdvDfltRt: Yes
Query Neighbor cache for fe80::206:2aff:fe66:c800
  IntfName: OSAQDIO46          IntfType: IPAQENET6
  LinkLayerAddr: 00062A66C800  State: Stale
  Type: Route                  AdvDfltRt: Yes
Query Neighbor cache for fe80::6000:1ff:feaa:e5a
     IntfName: OSXC9INT2          IntfType: IPAQENET6
     LinkLayerAddr: 620001AA0E5A  State: Reachable
     Type: Host                   AdvDfltRtr: No
Query Neighbor cache for 2001:db8:172::16:0:2
     IntfName: OSXC9INT2          IntfType: IPAQENET6
     LinkLayerAddr: 620001AA0E5A  State: Reachable
     Type: Host                   AdvDfltRtr: No
Query Neighbor cache for 2001:db8:172::16:0:2
     IntfName: EZ6IQXC9           IntfType: IPAQIQDX6    OSX: OSXC9INT2
     LinkLayerAddr: 820002AA0E22  State: Reachable
     Type: Host                   AdvDfltRtr: No
```

Report field descriptions:

Neighbor's IP address

IntfName
Interface name where the neighbor cache entry exists.

IntfType
Interface type.

OSX For HiperSockets interfaces that use the Internal Queued Direct I/O extensions function (IQDX), this field indicates the associated OSX interface.

LinkLayerAddr
Neighbor's link layer address (MAC address).

State Reachability state of the neighbor as defined in RFC 2461. Possible values include:

- **Incomplete**
  Address resolution has not been completed.

- **Reachable**
  Confirmation of neighbor's reachability received recently (within ReachableTime as defined by RFC 2461).

- **Stale** Reachability confirmation not recent.

- **Delay** Reconfirmation of reachability can be done after a short delay.

- **Probe** In process of reconfirming neighbor's reachability.

Type Neighbor type is either Host or Router.

AdvDfltRtr
Whether the neighbor advertised itself as a default router.

- **Y** Indicates the neighbor advertised itself as a default router.

- **N** Indicates the neighbor did not advertise itself as a default router.

Netstat PORTList/-o report

Displays the list of reserved ports and the port access control configuration for unreserved ports. To configure port access control for unreserved ports, replace the port number value with the keyword UNRSV. For more information about port access control see the port access control information in z/OS Communications Server: IP Configuration Guide. For ports that are reserved by the PORTRANGE profile statement, only one output line is displayed for each range. For ephemeral port specifications, see the Netstat CONFIG/-f report description.

TSO syntax:

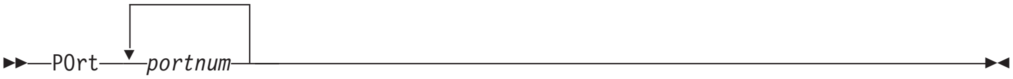
```
►►──NETSTAT PORTList──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

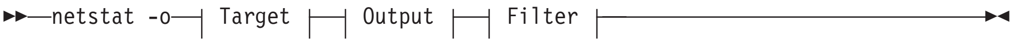
Output:

The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:



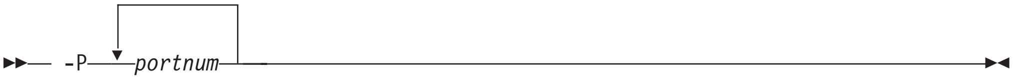
z/OS UNIX syntax:



Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Filter:



Filter description:

POrt/-P *portnum*
: Filter the output of the PORTList/**-O** report using the specified port number *portnum* or the keyword UNRSV. You can enter up to six filter values. The port number range is 1-65535.

Restriction: For a UDP endpoint socket, the filter value only applies to the local or source port.

Command syntax examples:

From TSO environment:

```
NETSTAT PORTLIST
Display the port reservation list in the default TCP/IP stack.
NETSTAT PORTLIST TCP TCPCS6
Display the port reservation list in the TCPCS6 stack.
```

From UNIX shell environment:

```
   netstat -o
   netstat -o -p tcpcs6
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT PORTLIST
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:24:23
Port# Prot User     Flags    Range       IP Address      SAF Name
----- ---- ----     -----    -----       ----------      --------
UNRSV TCP  A*       L
UNRSV TCP  *        FL                                   GENERIC
00020 TCP  FTPD1    D
00021 TCP  FTPD1    DA
00023 TCP  TCPCS    DA
00025 TCP  SMTP     DA
04000 TCP  OMVS     DABU                 9.67.113.10
04001 TCP  OMVS     DABFU                9.67.113.12     BS4TOMVS
04004 TCP  *        DAF                                  S4TALL
04005 TCP  *        DABU                 9.67.113.11
04017 TCP  *        DABFU                9.67.113.17     BS4TALL
04020 TCP  DCICSTS  DAN
05000 TCP  *        DARN     05000-05001
UNRSV UDP  *        XI
00161 UDP  OSNMPD   DA
00162 UDP  OMVS     DA
00514 UDP  SYSLOGD1 DA
04020 UDP  OMVS     DABF                 9.67.43.70      BS4UOMVS
04030 UDP  *        DAF                                  S4UALL
05000 UDP  MUD      DAR      05000-05002
```

IPv6 enabled or request for LONG format:

```
NETSTAT PORTLIST
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:24:23
Port# Prot User     Flags    Range       SAF Name
----- ---- ----     -----    -----       --------
UNRSV TCP  *        FL                   GENERIC
00020 TCP  FTPD1    D
00021 TCP  FTPD1    DA
00023 TCP  TCPCS    DA
00025 TCP  SMTP     DA
04000 TCP  OMVS     DABU
      BindSpecific: 9.67.113.10
04001 TCP  OMVS     DABFU                BS4TOMVS
      BINDSPECIFIC: 9.67.113.12
04002 TCP  OMVS     DABU
      BindSpecific: ::6:2900:1dc:21bc
04020 TCP  DCICSTS  DAN
05000 TCP  *        DARN     05000-05001
UNRSV UDP  *        FI                   GENERIC
00514 UDP  SYSLOGD1 DA
04020 UDP  OMVS     DAB
      BindSpecific: 9.67.43.70
04022 UDP  *        DAB
      BindSpecific: 1::8
04030 UDP  *        DA
05000 UDP  MUD      DAR      05000-05002
```

Report field descriptions:
Display the following port reservation information defined in the PORT or PORTRANGE profile statements. For more information about each field, see the PORT or PORTRANGE profile statements in the z/OS Communications Server: IP Configuration Reference.

Port#

nnnn For ports reserved by the PORT profile statement, this value is the number of the port that was reserved. For ports that are reserved by the PORTRANGE profile statement, this value is the number of the first port in the range. Valid values are in the range 1 – 65535.

UNRSV
Indicates any unreserved port; that is, any port number in the range 1-65535 that has not been reserved by a PORT or PORTRANGE statement. For applications that explicitly bind to an unreserved port and match the protocol and *jobname* value on this PORT statement, permission to access the unreserved port is controlled according to the value of the flags for that entry. However, when the RESTRICTLOWPORTS parameter is configured on the TCPCONFIG or UDPCONFIG profile statement, access only to unreserved ports with port numbers greater than 1023 is controlled by the PORT UNRSV statements.

Prot The protocol that was specified in the PORT profile statement. The valid protocol values are TCP and UDP.

User The MVS job name that can use the port. See Client name or User ID descriptions in "Netstat report general concept" on page 324 for detailed descriptions.

Flags The flags represent parameter values defined on the PORT or PORTRANGE profile statement.

A Autolog

B Bind

D DelayAcks

F SAF

I WhenBind

L WhenListen

N Port is disabled for SMC-R. For more information about SMC-R support, see Shared Memory Communications over Remote Direct Memory Access in the z/OS Communications Server: IP Configuration Guide.

R Port is reserved by range.

S Share port

U Reuse port. This flag is set for TCP sockets when the BIND keyword is specified (both B and U are set).

W Shareport with WLM server-specific weights is being used.

X Deny

Range This field is significant only for port entry reserved by the PORTRANGE profile statement (flag R in the Flags field).

IP address or BindSpecific
This field is significant only for port entries with the BIND parameter specified on the PORT profile statement.

SAF Name
The final qualifier of a security product resource name.

Netstat RESCache/-q report

Displays system-wide resolver cache information. This information is not specific to the TCP/IP stack whose name was specified on the TCp/**-p** target parameter or to the default TCP/IP stack. Statistical information, such as number of record entries or number of cache queries, can be retrieved, or detailed information about some or all of the cache entries can be retrieved. Resolver caching is configured using resolver configuration statements in the resolver setup file. For more information about resolver caching, see details about resolver caching in z/OS Communications Server: IP Configuration Guide.

TSO syntax:

```
►►──NETSTAT RESCache──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├────────►◄
```

Modifier:

```
      ┌─SUMmary──────────────┐
►►────┼──────────────────────┼───────────────────────────────────────────────►◄
      ├─DETAIL──┬──────────┬─┤
      │         └─NEGative─┘ │
      └─SUMmary──┬─────┬─────┘
                 └─DNS─┘
```

DETAIL

Display detailed information for all unexpired entries that are currently in the resolver cache. It includes the following information:

- Host-name-to-IP-address entries from resolver forward lookups.
- IP-address-to-host-name entries from resolver reverse lookups.
- Negative entries that are included in both forward and reverse lookup tables.

NEGative

Display detailed information for all negative cache entries in the resolver cache.

SUMmary

Display general system statistics for resolver cache operations. This is the default report for the RESCACHE report option.

DNS Display general system statistics for resolver cache operations, plus individual statistics for each DNS name server that has provided information currently stored in the cache.

Target:
Provide the report for a specific TCP/IP address space by using the TCp *tcpname* option. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user terminal. For other options, see "Netstat command output" on page 316.

Filter:

```
►►──┬─DNSAddr──dnsaddr───┬─────────────────────────────────────────────────►◄
    ├─HOSTName──hostname─┤
    └─IPAddr──ipaddr─────┘
```

z/OS UNIX syntax:

```
►►──netstat -q──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────►◄
```

Modifier:

```
    ┌─SUMmary──────────────┐
►►──┼──────────────────────┼───────────────────────────────────────────────►◄
    ├─DETAIL──┬──────────┬─┤
    │         └─NEGative─┘ │
    └─SUMmary──┬─────┬─────┘
               └─DNS─┘
```

DETAIL

Display detailed information for all unexpired entries that are currently in the resolver cache. See the following entries:

- Host-name-to-IP-address entries from resolver forward lookups.
- IP-address-to-host-name entries from resolver reverse lookups.
- Negative entries that are included in both forward and reverse lookup tables.

NEGative

Display detailed information for all negative cache entries that are in the resolver cache.

SUMmary

Display general system statistics for resolver cache operations. This is the default report for the RESCACHE report option.

DNS Display general system statistics for resolver cache operations, plus individual statistics for each DNS name server that has provided information currently stored in the cache.

Target:

Provide the report for a specific TCP/IP address space by using the TCp *tcpname* option. See "The Netstat command target" on page 316 for more information about the **-p** parameter.

Ouptut:

The default output option displays the output on the user terminal. For other options, see "Netstat command output" on page 316.

Filter:

```
►►──┬─-H──hostname─┬───────────────────────────────────────────────────────►◄
    ├─-I──ipaddr───┤
    └─-Q──dnsaddr──┘
```

Filter description:

DNSAddr/-Q *dnsaddr*

Filter the output of the RESCache/**-q** report using the specified DNS IP address *dnsaddr*. You can enter one filter value at a time. The specified IPv4 *dnsaddr* value can be up to 15 characters in length; the specified IPv6 *dnsaddr* value can be up to 45 characters in length.

Restriction: The filter value does not support wildcard characters.

HOSTName/-H *hostname*

Filter the output of the RESCache/**-q** report using the specified host name value *hostname*. You can enter one filter value at a time. The specified value can be up to 255 characters in length.

Restriction: The HOSTName/**-H** filter applies only to the `IPAddress to HostName translation` portion of the report

The filter value for HOSTName/**-H** can be a complete string or a partial string that can use wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, in the same position. A wildcard character can also be a question mark (?), which matches any single character in the same position. For example, the string *searchee* matches with the search value *ar?he*, but the string *searhee* does not match with the search value *ar?he*.

In addition to the asterisk (*) and question mark (?) wildcard characters, the HOSTName/**-H** filter also supports implicit wildcards. Implicit wildcards are handled similarly to the way resolver queries handle host names. For example, if you specify the filter value `host21`, the query returns resolver cache information for the following cache entries:

- host21
- host21.ibm.com
- host21.raleigh.ibm.com
- Any entry that has host21 as the value that precedes the first period in the fully-qualified domain name

However, if you specify the filter value `host21.`,the query matches only the entry `host21`.

Tip: The filter value `host21` is equal to `host21.*` but `host21` is not equal to `host21*` because `host21*` also matches cache entries such as host211.ibm.com and host2134.ibm.com.

When you use the z/OS UNIX **netstat/onetstat** command in a z/OS UNIX shell environment, using a z/OS UNIX MVS special character in a character string might cause an unpredictable result. If you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single quotation marks (') quotation marks ("). For example, to use an asterisk (*) in the host name, cat.* for the **-H** filter, issue the command as: **netstat -q -H 'cat.*'** or **netstat -q -H "cat.*"**.

IPAddr/-I *ipaddr*

Filter the output of the RESCache/**-q** report using the specified IP address *ipaddr*. You can enter one filter value at a time. The specified IPv4 *ipaddr* value can be up to 15 characters in length; the specified IPv6 *ipaddr* value can be up to 45 characters in length.

Restrictions:

- The filter value does not support wildcard characters.

- The IPAddr/**-I** filter applies only to the `IPAddress to HostName translation` portion of the report.

Use the RESCache/**-q** filters only when you use the DETAIL modifier. Specifying a filter with the SUMMARY modifier does not affect the report output.

Command syntax examples:

From TSO environment:

```
NETSTAT RESCACHE SUMMARY
  Display general system statistics for resolver cache operations
NETSTAT RESCACHE SUMMARY DNS
  Display general system statistics for resolver cache operations, plus individual
  statistics for each DNS name server that has provided information currently
  stored in the cache.
NETSTAT RESCACHE DETAIL
  Display detailed information for all unexpired entries that are currently
  in the resolver cache.
NETSTAT RESCACHE DETAIL (DNSADDR 10.7.7.7
  Display detailed information for all unexpired entries that are currently
  in the resolver cache that were provided by the DNS name server at IP address 10.7.7.7
NETSTAT RESCACHE DETAIL (HOSTName hostname.domain
  Display detailed information for all HostName to IPAddress resolution cache
  entries currently in the resolver cache that were acquired using host name
  hostname.domain as the target resource for the resolver query.
NETSTAT RESCACHE DETAIL (HOSTName hostname.*
  Display detailed information for all HostName to IPAddress resolution cache
  entries currently in the resolver cache that were acquired using a host name matching
  the hostname.* pattern string as the target resource for the resolver query.
NETSTAT RESCACHE DETAIL (IPAddr 10.9.9.9
  Display detailed information for all IPAddress to HostName resolution
  cache entries currently in the resolver cache that were acquired using IP address
  10.9.9.9 as the target resource for the resolver query.
NETSTAT RESCACHE DETAIL NEGATIVE
  Display detailed information for all negative cache entries in the resolver cache
```

From UNIX shell environment:

```
netstat -q SUMMARY
netstat -q SUMMARY DNS
netstat -q DETAIL
netstat -q DETAIL -Q 10.7.7.7
netstat -q DETAIL -H hostname.domain
netstat -q DETAIL -H 'hostname.*'
netstat -q DETAIL -I 10.9.9.9
netstat -q DETAIL NEGATIVE
```

Report examples:
The following examples are generated by using the TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

NETSTAT RESCACHE or **NETSTAT RESCACHE SUMMARY**

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:12:31
Storage Usage:
  Maximum: 10M
    Current: 203K   MaxUsed: 1M

Cache Usage:
  Total Number of entries: 64
    Non-NX entries: 44
      A: 20         AAAA: 13         PTR: 11
    NX entries: 20
      A: 9          AAAA: 2          PTR: 9
  Queries: 112                  Hits: 34
  SuccessRatio: 30%
```

NETSTAT RESCACHE SUMMARY DNS

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:12:45
Storage Usage:
  Maximum: 10M
    Current: 203K   MaxUsed: 1M

Cache Usage:
  Total Number of entries: 64
    Non-NX entries: 44
      A: 20         AAAA: 13         PTR: 11
    NX entries: 20
      A: 9          AAAA: 2          PTR: 9
  Queries: 112                  Hits: 34
  SuccessRatio: 30%

DNS address: 19.47.135.295
  Total Number of entries: 54
    Non-NX entries: 39
      A: 18         AAAA: 11         PTR: 10
    NX entries: 15
      A: 7          AAAA: 2          PTR: 6
  References: 77                   Hits: 21

DNS address: 19.52.206.22
  Total Number of entries: 10
    Non-NX entries: 5
      A: 2          AAAA: 2          PTR: 1
    NX entries: 5
      A: 2          AAAA: 0          PTR: 3
  References: 43                   Hits: 13
```

NETSTAT RESCACHE DETAIL

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:12:51
HostName to IPAddress translation
---------------------------------
HostName: HOSTNAME1
  DNS IPAddress: 19.47.135.295
  DNS Record Type: T_A
  Canonical Name: ***NA***
  Cache Time: 10/27/2011 18:44:43
  Expired Time: 10/27/2011 19:44:33
  Hits: 0
  IPAddress: ***NA***

HostName: HOSTNAME4.TCP.RALEIGH.IBM.COM
  DNS IPAddress: 19.47.135.295
  DNS Record Type: T_A
  Canonical Name: hostname4.tcp.raleigh.ibm.com
  Cache Time: 10/27/2011 18:36:51
  Expired Time: 10/28/2011 14:25:16
  Hits: 0
  IPAddress: 29.72.105.195
             29.72.105.196
             29.72.105.197

HostName: HOSTNAME1.TCP.RALEIGH.IBM.COM
  DNS IPAddress: 19.47.135.295
  DNS Record Type: T_A
  Canonical Name: ***NA***
  Cache Time: 10/27/2011 18:44:38
  Expired Time: 10/27/2011 19:44:28
  Hits: 0
  IPAddress: ***NA***

HostName: HOSTNAME5.TCP.RALEIGH.IBM.COM
  DNS IPAddress: 19.47.135.295
  DNS Record Type: T_A
  Canonical Name: hostname5.pok.ibm.com
  Cache Time: 10/27/2011 18:43:22
  Expired Time: 10/27/2011 20:09:38
  Hits: 0
  IPAddress: 29.236.231.69

HostName: WWW.NEWSPAPER.COM
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_A
  Canonical Name: newspaper.com
  Cache Time: 10/27/2011 19:05:29
  Expired Time: 10/27/2011 20:05:29
  Hits: 0
  IPAddress: 125.152.8.134
```

```
HostName: HOSTNAME5.TCP.RALEIGH.IBM.COM
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_A
  Canonical Name: hostname55.pok.ibm.com
  Cache Time: 10/27/2011 19:06:36
  Expired Time: 10/27/2011 20:09:37
  Hits: 1
  IPAddress: 29.236.231.69

HostName: WWW.STATE1.GOV
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_AAAA
  Canonical Name: state1.gov
  Cache Time: 10/27/2011 19:09:24
  Expired Time: 10/27/2011 19:19:24
  Hits: 0
  IPAddress: 144::227:12:34:76

HostName: WWW.COLLEGE1.EDU
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_AAAA
  Canonical Name: www.college1.edu
  Cache Time: 10/27/2011 19:08:45
  Expired Time: 10/27/2011 19:18:45
  Hits: 0
  IPAddress: 1004::251:133:180:120
             1004::251:133:180:121

HostName: WWW.COMPANY.COM
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_AAAA
  Canonical Name: ***NA***
  Cache Time: 10/27/2011 19:11:20
  Expired Time: 10/27/2011 19:28:00
  Hits: 2
  IPAddress: ***NA***
```

```
IPAddress to HostName translation
---------------------------------
IPAddress: 930::159:84:218:223
  DNS IPAddress: 19.47.135.295
  DNS Record Type: T_PTR
  Cache Time: 10/27/2011 19:07:43
  Expired Time: 10/27/2011 19:17:43
  Hits: 1
  HostName: hostipv6-223.218.84.150.company.com

IPAddress: 152.12.39.164
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_PTR
  Cache Time: 10/27/2011 19:05:59
  Expired Time: 10/27/2011 20:05:59
  Hits: 1
  HostName: ***NA***

IPAddress: 159.84.218.223
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_PTR
  Cache Time: 10/27/2011 19:05:43
  Expired Time: 10/27/2011 19:15:43
  Hits: 1
  HostName: hostvalue-223.218.84.159.company.com

IPAddress: 152.152.31.134
  DNS IPAddress: 19.52.206.22
  DNS Record Type: T_PTR
  Cache Time: 10/27/2011 19:05:59
  Expired Time: 10/27/2011 20:05:59
  Hits: 1
  HostName: namemh.media.com
```

Report field descriptions:

SUMMARY or SUMMARY DNS reports:

Storage Usage
Displays information about the overall storage usage for resolver caching. This storage represents 64-bit private storage in the resolver address space.

Maximum
Displays the maximum amount of storage that the resolver can allocate to manage cache records. This value was defined using the CACHESIZE resolver setup statement. This value can be displayed as a number followed by the letter *M*.

Current
Displays the current amount of storage that the resolver has allocated to manage cache records. This value can be displayed in the following forms:

- If the value is less than 1 MB, then it is displayed as a number followed by the letter *K*.
- If the value is 1 MB or greater, it is displayed as a number followed by the letter *M*.

MaxUsed
Displays the greatest amount of storage that the resolver has ever allocated for managing cache records. This value can be displayed in the following forms:

- If the value is less than 1 MB, then it is displayed as a number followed by the letter *K*.
- If the value is 1 MB or greater, it is displayed as a number followed by the letter *M*.

Cache Usage
Displays information about the number and the makeup of the records that are currently in the resolver cache. This value includes both NX (negative cache entries) and non-NX entries.

Total number of entries
Displays the total number of A, AAAA, and PTR cache entries that are currently in the resolver cache. This value includes both NX (negative cache entries) and non-NX entries.

Non-NX entries
Displays the total number of A, AAAA, and PTR cache entries that represent successful name or address resolution attempts, that are in the resolver cache. An individual cache entry represents one of the following case:

- A host-name-to-IPv4-address resolver query to a specific name server, for which the name server returned at least one IPv4 address.
- A host-name-to-IPv6-address resolver query to a specific name server, for which the name server returned at least one IPv6 address.
- An IP-address-to-host-name resolver query to a specific name server, for which the name server returned a host name.

A Displays the total number of A cache entries that are currently in the resolver cache that were created as a result of a successful resolution attempt. An individual A cache entry contains the results of one host-name-to-IPv4-address resolver query to a specific name server, even if multiple IPv4 addresses were returned by the name server for the target host name.

AAAA
Displays the total number of AAAA cache entries that are currently in the resolver cache that were created as a result of a successful resolution attempt. An individual AAAA cache entry contains the results of one host-name-to-IPv6-address resolver query to a specific name server, even if multiple IPv6 addresses were returned by the name server for the target host name.

PTR
Displays the total number of PTR cache entries that are currently in the resolver cache that were created as a result of a successful resolution attempt. An individual PTR cache entry contains the results of one IP-address-to-host-name resolver query to a specific name server. The target IP address can be either an IPv4 address or an IPv6 address.

NX entries
Displays the total number of A, AAAA, and PTR negative cache entries that are in the resolver cache. An individual negative cache entry represents one of the following results:

- A host-name-to-IPv4-address resolver query to a specific name server, in which the name server indicated that the target host name does not have associated IPv4 addresses.
- A host-name-to-IPv6-address resolver query to a specific name server, in which the name server indicated that the target host name does not have associated IPv6 addresses.
- An IP-address-to-host-name resolver query to a specific name server, in which the name server indicated that the target IP address does not have associated host name value. The IP address can be either an IPv4 address or an IPv6 address.

A Displays the total number of A negative cache entries that are currently in the resolver cache that were created as a result of host-name-to-IPv4-address resolver queries that did not return any IPv4 addresses. An individual A negative cache entry contains the results of one host-name-to-IPv4-address resolver query to a specific name server.

AAAA
Displays the total number of AAAA negative cache entries that are currently in the resolver cache that were created as a result of host-name-to-IPv6 address resolver queries that did not return any IPv6 addresses. An individual AAAA negative cache entry contains the results of one host-name-to-IPv6-address resolver query to a specific name server.

PTR
Displays the total number of PTR negative cache entries that are currently in the resolver cache that were created as a result of IP-address-to-host-name resolver queries that did not return a host name. An individual PTR negative cache entry contains the results of one IP-address-to-host-name resolver query to a specific name server. The target IP address can be either an IPv4 address or an IPv6 address.

Queries

Displays the total number of instances in which a query to the resolver cache services was attempted.

Guideline: A single resolver API call, for example, Getaddrinfo, might result in multiple queries to the resolver cache. Separate cache queries are attempted for IPv4 and IPv6 information. Separate cache queries are also attempted for different domains if the SEARCH resolver configuration statement specifies that different domains should be appended to the input host name value for search purposes.

Hits

Displays the total number of instances in which a query of the resolver cache provided response information about the target resource. The response information represents either DNS reply information that had been cached, or it represents an indication that negative cache information existed for the target resource.

SuccessRatio

Displays the percentage of cache queries that successfully provided response information. This value is equal to the number of hits divided by the number of queries.

DNS address

Displays information about the number and the makeup of the records that are currently in the resolver cache that are associated with replies from the name server at the displayed IP address.

Total number of entries

Displays the total number of A, AAAA, and PTR cache entries that are currently in the resolver cache that were created as a result of DNS response information from this specific name server. This value includes both NX (negative cache) entries and non-NX entries.

Non-NX entries

Displays the total number of A, AAAA, and PTR cache entries that represent successful name or address resolution attempts, in the resolver cache as a result of DNS response information from this specific name server. An individual cache entry contains one of the following results:

- A host-name-to-IPv4-address resolver query to a specific name server, for which the name server returned at least one IPv4 address.
- A host-name-to-IPv6-address resolver query to a specific name server, for which the name server returned at least one IPv6 address.
- An IP-address-to-host-name resolver query to a specific name server, for which the name server returned a host name.

A

Displays the total number of A cache entries that are currently in the resolver cache that were created as a result of a successful resolution attempt directed to this specific name server. An individual A cache entry contains the results of one host-name-to-IPv4-address resolver query, even if multiple IPv4 addresses were returned by the name server for the target host name.

AAAA

Displays the total number of AAAA cache entries that are currently in the resolver cache that were created as a result of a successful resolution attempt directed to this specific name server. An individual AAAA cache entry contains the results of one host-name-to-IPv6-address resolver query, even if multiple IPv6 addresses were returned by the name server for the target host name.

PTR

Displays the total number of PTR cache entries that are currently in the resolver cache that were created as a result of a successful resolution attempt directed to this specific name server. An individual PTR cache entry contains the results of one IP-address-to-host-name resolver query. The target IP address can be either an IPv4 address or an IPv6 address.

NX entries

Displays the total number of A, AAAA, and PTR negative cache entries in the resolver cache that were created as a result of DNS response information from this specific name server. An individual negative cache entry can contain any of the following results:

- A host-name-to-IPv4-address resolver query to which the name server responded that the target host name has no associated IPv4 addresses.
- A host-name-to-IPv6-address resolver query to which the name server responded that the target host name has no associated IPv6 addresses.
- An IP-address-to-host-name resolver query to which the name server responded that the target IP address has no associated host name value. The IP address can be either IPv4 addresses or IPv6 addresses.

A Displays the total number of A negative cache entries that are currently in the resolver cache that were created a result of host-name-to-IPv4-address resolver queries to this specific name server that did not return any IPv4 addresses. An individual A negative cache entry contains the results of one host-name-to-IPv4-address resolver query.

AAAA

Displays the total number of AAAA negative cache entries that are currently in the resolver cache that were created as a result of host-name-to-IPv6-address resolver queries to this specific name server that did not return any IPv6 addresses. An individual AAAA negative cache entry contains the results of one host-name-to-IPv6-address resolver query.

PTR

Displays the total number of PTR negative cache entries that are currently in the resolver cache that were created as a result of IP-address-to-host-name resolver queries to this specific name server that did not return any host name. An individual PTR negative cache entry contains the results of one IP-address-to-host-name resolver query. The target IP address can be either an IPv4 address or an IPv6 address.

References

Displays the total number of instances in which a query of the resolver cache examined the collection of cache entries that represents DNS response information from the name server that was specified by the DNS address in an attempt to find response information about the target resource.

Guideline: A single cache query attempt might result in multiple queries of different collections of information provided by the name server. You can use the NSINTERADDR resolver configuration statement to specify which name servers should be queried to obtain information about the target resource, and in which order they should be queried. The resolver cache uses the same NSINTERADDR list to determine which collections of information provided by the name server to examine for a specific cache query. If no response information is found in the collection of information that is provided by the first name server in the NSINTERADDR list, the collection of information that is provided by the second name server in the list is checked, and so on.

If the first name server in the NSINTERADDR list is the primary name server for the installation, the number of references for that name server should be much higher than the number of references for the other name servers in the list. Most of the resolver queries should, during normal operation, be directed to the primary name server, and therefore most information in the resolver cache should be provided by the primary name server. A comparatively small number of references for the primary name server, compared to the secondary name servers, suggests that the primary name server might not be active all the time, or that most of the cache queries are not successful, because the resolver cache is examining more than just the primary name server's collection of responses as part of a single cache query.

Hits

Displays the total number of instances in which a query of the resolver cache provided response information about the target resource that was created as a result of DNS response information from the name server that was specified by the DNS address. The response information representss either DNS reply information that had been cached, or it is an indication that negative cache information existed for the target resource.

DETAIL or DETAIL NEGATIVE reports:

HostName to IPAddress translation

This banner indicates that the next set of cache entries that is being reported contains the results of resolver queries, such as Getaddrinfo and Gethostbyname, to translate a host name into one or more IP addresses.

HostName

Displays the host name that is used as the target resource to acquire the cache information that is in this entry.

DNS IPAddress

Displays the IP address of the name server that provided the response information about the host name that is contained in this cache entry.

DNS Record Type

Displays the record type of this cache entry. See the following possible values:

T_A

Indicates the result of an attempt to resolve a host name to an IPv4 address.

T_AAAA
Indicates the result of an attempt to resolve a host name to an IPv6 address.

Canonical Name
Displays the official DNS name for the host name. The name is provided as part of the DNS response in the form of a T_CNAME resource record. If this entry represents negative cache information, then the value `*** NA ***` is displayed.

Cache Time
Displays the time and date when the cache entry was created.

Expired Time
Displays the time and date after which the cache entry is no longer valid. A time-to-live (TTL) value is provided as part of the DNS response data, and that value indicates how long the response data can be trusted. You can use the resolver MAXTTL setup statement to change the TTL value that is returned by DNS. See MAXTTL statement information in z/OS Communications Server: IP Configuration Reference for details about the resolver MAXTTL setup statement.

Guideline: The resolver cache logic does not automatically delete cache entries when the TTL time expires. The cache entries are deleted when a subsequent query for the host name is received or they are deleted as part of periodic storage cleanup processing.

Hits
Displays the number of times that the information in this cache entry was used to respond to a cache query.

IPAddress
Displays one or more IP addresses that are associated with the host name. One cache entry is used to contain IPv4 addresses that are associated with a given host name, and a second entry is used to contain IPv6 addresses that are associated with a given host name. All of the addresses displayed for a host name are either all IPv4 addresses, or all IPv6 addresses, but never a mixture of both types of addresses. If this entry represents negative cache information, then the value `*** NA ***` is displayed instead of an IP address.

IPAddress to HostName translation
This banner indicates that the next set of cache entries that is being reported displays the results of resolver queries, such as Getnameinfo and Gethostbyaddr, to translate an IP address into a host name.

IPAddress
Displays the IP address that is used as the target resource to acquire the cache information in this entry.

DNS IPAddress
Displays the IP address of the name server that provided the response information about the IP address that is contained in this cache entry.

DNS Record Type
Displays the record type of this cache entry. The possible value is:

T_PTR
Indicates the result of an attempt to resolve an IP address to a host name.

`Cache Time`
: Displays the time and date when the cache entry was created.

`Expired Time`
: Displays the time and date when the cache entry is longer valid. A time-to-live (TTL) value is provided as part of the DNS response data; that value indicates how long the response data can be trusted. You can use the resolver MAXTTL setup statement to change the TTL value that is returned by DNS. See the MAXTTL statement information in z/OS Communications Server: IP Configuration Reference for details about the resolver MAXTTL setup statement.

  Guideline: The resolver cache logic does not automatically delete cache entries when the TTL time expires. The cache entries are deleted when a subsequent query for the IP address is received, or they are deleted as part of periodic storage cleanup processing.

`Hits`
: Displays the number of times that the information in this cache entry was used to respond to a cache query.

`HostName`
: Displays the host name that is associated with the IP address.

  If this entry represents negative cache information, then the value `*** NA ***` is displayed instead of an IP address.

Netstat ROUTe/-r report

Displays the routing information that this stack uses when it determines what addresses it can communicate with and over which links or interfaces and first hops the communication takes place. The routes in the stack main routing table can be displayed, as well as the routes in the stack policy-based routing tables. These routes can be static routes (those defined in the TCP/IP profile for the main route table and those defined to the Policy Agent for policy-based route tables), routes learned from routing daemons, and routes learned by other ICMP or ICMPv6 information, such as redirects. If there is no route that covers the destination IP address and if there is no default route defined, then this stack cannot communicate with that destination. Multiple routes to the same destination, referred to as multipath routes, are also displayed. If multipath is not enabled (on the IPCONFIG or IPCONFIG6 statement for the main route table and on the RouteTable policy statement for policy-based route tables), then the first active route to the destination is always used.

Tip: Static routes over deleted interfaces are removed from the main routing table and therefore do not appear in reports that are generated for the main routing table. Loopback routes are displayed as well as implicit (HOME list) routes.

TSO syntax:

```
►►──NETSTAT ROUTe──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────►◄
```

Modifier:

```
        ┌────────────────────────────┐
        ▼                            │
►►──────┬──────────────────────────┬─┴──────────────────────────────────────────►◄
        ├─ADDRTYPE───┬─IPV4─┬──────┤
        │            └─IPV6─┘      │
        ├─DETAIL───────────────────┤
        ├─IQDIO────────────────────┤
        ├─PR───┬─ALL────┬──────────┤
        │      └─prname─┘          │
        ├─QDIOACCEL────────────────┤
        ├─RADV─────────────────────┤
        └─RSTAT────────────────────┘
```

ADDRTYPE IPV4 | IPV6

Display the specified IP type routing information.

IPV4 Display IPv4 routing information. This parameter is mutually exclusive with the RADV parameter.

IPV6 Display IPv6 routing information.

DETAIL

Displays additional details such as the metric or cost of use for the route, MTU size if it is an IPv4 route, and the MVS specific configured parameters for each route.

This parameter is mutually exclusive with the QDIOACCEL and IQDIO parameters.

PR Displays policy-based routing tables. This parameter is mutually exclusive with the QDIOACCEL and IQDIO parameters.

ALL Displays all policy-based routing tables.

prname

Displays the policy-based routing table that has the name *prname*.

Restriction: The Netstat ROUTe command displays only active policy-based route tables. A policy-based route table is active if it is referenced by an active routing rule and its associated action. You can display active and inactive policy-based route tables with the **pasearch** command. For more information, see "The z/OS UNIX pasearch command: Display policies" on page 819.

QDIOACCEL
IQDIO

Displays the routes that are eligible for accelerated routing using QDIO Accelerator or HiperSockets Accelerator. See the QDIO Accelerator information and the information about efficient routing using HiperSockets Accelerator in z/OS Communications Server: IP Configuration Guide for more details. This parameter is mutually exclusive with the DETAIL, RADV, PR, and RSTAT parameters.

RADV

Displays all of the IPv6 routes that are added based on information received in router advertisement messages. All IPv6 router advertisement routes are displayed whether or not they are currently used for routing. The flags and reference count are not displayed on the report. This parameter is mutually exclusive with the RSTAT, QDIOACCEL, IQDIO, and ADDRTYPE IPV4 parameters.

RSTAT

Displays all of the static routes that are defined as replaceable. All defined replaceable static routes are displayed whether or not they are currently being used for routing. The flags and reference count are not displayed on the report. This parameter is mutually exclusive with the RADV, QDIOACCEL, and IQDIO parameters.

Target:

Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
                 ┌─────────────────────┐
                 ▼                     │
►►──IPAddr──────┬─ipaddr───────────────┴─────────────────────────────────►◄
                ├─ipaddr/prefixLen──┤
                └─ipaddr/subnetmask─┘
```

z/OS UNIX syntax:

```
►►──netstat -r──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────►◄
```

Modifier:

```
      ┌──────────────────────┐
      ▼                      │
►►──────┬─ADDRTYPE──┬─IPV4─┬─┬┴──────────────────────────────────────────►◄
        │           └─IPV6─┘ │
        ├─DETAIL─────────────┤
        ├─IQDIO──────────────┤
        ├─PR───┬─ALL────┬────┤
        │      └─prname─┘    │
        ├─QDIOACCEL──────────┤
        ├─RADV───────────────┤
        └─RSTAT──────────────┘
```

ADDRTYPE IPV4 | IPV6

Display the specified IP type routing information.

IPV4 Display IPv4 routing information. This parameter is mutually exclusive with the RADV parameter.

IPV6 Display IPv6 routing information.

DETAIL

Displays additional details such as the metric or cost of use for the route, MTU size if it is an IPv4 route, and the MVS-specific configured parameters for each route. This parameter is mutually exclusive with the QDIOACCEL and IQDIO parameters.

PR Displays policy-based routing tables. This parameter is mutually exclusive with the QDIOACCEL and IQDIO parameters.

ALL Displays all policy-based routing tables.

prname
Displays the policy-based routing table that has the name *prname*.

Restriction: The Netstat ROUTe command displays only active policy-based route tables. A policy-based route table is active if it is referenced by an active routing rule and its associated action. You can display active and inactive policy-based route tables with the **pasearch** command. For more information, see "The z/OS UNIX pasearch command: Display policies" on page 819.

QDIOACCEL
IQDIO
Displays the routes that are eligible for accelerated routing using QDIO Accelerator or HiperSockets Accelerator. See QDIO Accelerator information and efficient routing using HiperSockets Accelerator information in z/OS Communications Server: IP Configuration Guide for more details. This parameter is mutually exclusive with the DETAIL, RADV, PR, and RSTAT parameters.

RADV
Displays all of the IPv6 routes that are added based on information received in router advertisement messages. All IPv6 router advertisement routes are displayed whether or not they are currently used for routing. The flags and reference count are not displayed on the report. This parameter is mutually exclusive with the RSTAT, QDIOACCEL, IQDIO, and ADDRTYPE IPV4 parameters.

RSTAT
Displays all static routes that are defined as replaceable. All defined replaceable static routes are displayed without regard to whether or not they are currently being used for routing. The flags and reference count are not displayed on the report. This parameter is mutually exclusive with the RADV, QDIOACCEL, and IQDIO parameters.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
              ┌─────────────────────┐
              ▼                     │
►►──-I────┬─ipaddr──────────────┬───┴──────────────────────────────────────►◄
          ├─ipaddr/prefixLen────┤
          └─ipaddr/subnetmask───┘
```

Filter description:

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr
Filter the output of the ROUTe/**-r** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength
Filter the output of the ROUTe/**-r** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask
Filter the output of the ROUTe/**-r** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

The IPAddr/**-I** filter value can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/**-I** filter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/prefixlen* or *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -r -I '10.*.0.0'** or **netstat -r -I "10.*.0.0"**.

Note: When filtering ROUTe/**-r** responses on a specified IP address, the DEFAULT and DEFAULTNET routes are not displayed.

Guidelines:

1. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.

- For an IPv6-enabled stack, an IPv4-mapped IPv6 address is accepted and is treated as an IPv6 address. If an IPv4-mapped IPv6 address is entered as an IPAddr/**-I** value, there is no matching entry found.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.

Command syntax examples:

From TSO environment:

```
NETSTAT ROUTE
   Display the routing information the default stack will use when it determines what
   addresses it can communicate with and over which links/interfaces and first hops the
   communication will take place. If the stack is IPv6-enabled, then both IPv4 and IPv6
   routing information are displayed.
NETSTAT ROUTE TCP TCPCS6
   Display the routing information the TCPCS6 stack will use when it determines what
   addresses it can communicate with and over which links/interfaces and first hops the
   communication will take place. If the TCPCS6 stack is IPv6-enabled, then both IPv4 and
   IPv6 routing information are displayed.
NETSTAT ROUTE TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display the routing information in the TCPCS8 stack whose destination address match
   the specified filter IP address values.
NETSTAT ROUTE ADDRTYPE IPV4
   Display the IPv4 routing information the default stack will use when it determines
   what addresses it can communicate with and over which links/interfaces and first hops
   the communication will take place.
```

From UNIX shell environment:

```
   netstat -r
   netstat -r -p tcpcs6
   netstat -r -p tcpcs8 -I 9.43.1.1 9.43.2.2
   netstat -r ADDRTYPE IPV4
```

Report examples:

The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT ROUTE or
NETSTAT ROUTE ADDRTYPE IPV4

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:24:09
Destination        Gateway         Flags    Refcnt     Interface
-----------        -------         -----    ------     ---------
Default            9.67.115.65     UGS      0000000002 OSAQDIOLINK
9.67.115.65/32     0.0.0.0         UHS      0000000000 OSAQDIOLINK
9.67.115.69/32     0.0.0.0         UH       0000000000 OSAQDIOLINK
127.0.0.1/32       0.0.0.0         UH       0000000002 LOOPBACK
```

```
NETSTAT ROUTE DETAIL

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:03:13
Destination        Gateway         Flags    Refcnt     Interface
-----------        -------         -----    ------     ---------
Default            9.67.115.1      UGS      0000000000 OSAQDIO5L
  Metric: 00000001  MTU: 1496
  MVS Specific Configured Parameters:
    MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
    RoundTripGain:      0.125     VarianceGain:      0.250
    VarianceMultiplier: 2.000     DelayAcks:         Yes
```

```
NETSTAT ROUTE RSTAT

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:40:36
IPv4 Destinations
Destination        Gateway         Interface
-----------        -------         ---------
9.67.1.9/32        0.0.0.0         OSA00LINK1
```

```
NETSTAT ROUTE QDIOACCEL

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            09:51:02
Destination        Gateway         Interface
-----------        -------         ---------
9.67.1.9/32        0.0.0.0         LIQDIO1
9.67.5.10/32       0.0.0.0         OSAQDIO5L
```

```
NETSTAT ROUTE PR prtable1

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:24:09
Policy Routing Table: prtable1
  IgnorePathMtuUpdate: Yes  MultiPath: Conn(Policy)
  DynamicXCFRoutes:    No
Dynamic Routing Parameters
  Interface        NextHop
  ---------        -------
  OSAQDIOLINK      9.67.115.65
Destination        Gateway         Flags    Refcnt     Interface
-----------        -------         -----    ------     ---------
Default            9.67.115.65     UGS      0000000002 OSAQDIOLINK
9.67.115.65/32     0.0.0.0         UHS      0000000000 OSAQDIOLINK
9.67.115.69/32     0.0.0.0         UH       0000000000 OSAQDIOLINK
```

IPv6 enabled or request for LONG format:

```
NETSTAT ROUTE

MVS TCP/IP NETSTAT CS V2R1      TCPIP Name: TCPCS           14:24:09
IPv4 Destinations
Destination Gateway Flags Refcnt Interface
-        -         -    -     -
Default            9.67.115.65     UGS      0000000002 OSAQDIOLINK
9.67.115.65/32     0.0.0.0         UHS      0000000000 OSAQDIOLINK
9.67.115.69/32     0.0.0.0         UH       0000000000 OSAQDIOLINK
127.0.0.1/32       0.0.0.0         UH       0000000002 LOOPBACK
IPv6 Destinations
DestIP:   Default
  Gw:     2001:0db8::206:2aff:fe71:4400
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UGS               MTU:     1492
DestIP:   ::1/128
  Gw:     ::
  Intf:   LOOPBACK6         Refcnt:  0000000000
  Flgs:   UH                MTU:     65535
DestIP:   2001:0db8::9:67:115:13/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UD                MTU:     1492
DestIP:   2001:0db8::206:2aff:fe71:4400/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UHS               MTU:     1492
```

```
NETSTAT ROUTE ADDRTYPE IPV4

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:24:09
IPv4 Destinations
Destination        Gateway         Flags    Refcnt     Interface
-----------        -------         -----    ------     ---------
Default            9.67.115.65     UGS      0000000002 OSAQDIOLINK
9.67.115.65/32     0.0.0.0         UHS      0000000000 OSAQDIOLINK
9.67.115.69/32     0.0.0.0         UH       0000000000 OSAQDIOLINK
127.0.0.1/32       0.0.0.0         UH       0000000002 LOOPBACK
```

```
NETSTAT ROUTE ADDRTYPE IPV6

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:24:09
IPv6 Destinations
DestIP:   Default
  Gw:     2001:0db8::206:2aff:fe71:4400
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UGS               MTU:     1492
DestIP:   ::1/128
  Gw:     ::
  Intf:   LOOPBACK6         Refcnt:  0000000000
  Flgs:   UH                MTU:     65535
DestIP:   2001:0db8::9:67:115:13/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UD                MTU:     1492
DestIP:   2001:0db8::206:2aff:fe71:4400/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UHS               MTU:     1492
```

NETSTAT ROUTE DETAIL

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:03:13
IPv4 Destinations
Destination        Gateway         Flags    Refcnt     Interface
-----------        -------         -----    ------     ---------
Default            9.67.115.1      UGS      0000000000 OSAQDIO5L
  Metric: 00000001  MTU: 1496
  MVS Specific Configured Parameters:
    MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
    RoundTripGain:      0.125     VarianceGain:      0.250
    VarianceMultiplier: 2.000     DelayAcks:         Yes
......

IPv6 Destinations
......

DestIP:   2001:0db8::206:2aff:fe71:4400/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  0000000000
  Flgs:   UHS               MTU:     1492
  Metric: 00000000
  MVS Specific Configured Parameters:
    MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
    RoundTripGain:      0.125     VarianceGain:      0.250
    VarianceMultiplier: 2.000     DelayAcks:         Yes
```

NETSTAT ROUTE RSTAT

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:40:36
IPv4 Destinations
Destination        Gateway         Interface
-----------        -------         ---------
9.67.1.9/32        0.0.0.0         OSA00LINK1

IPv6 Destinations
DestIP:   fe80::6:2900:1dc:21bc/128
  Gw:     ::
  Intf:   OSAQDIO46  MTU:  1492
```

NETSTAT ROUTE QDIOACCEL

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            09:51:02
Destination        Gateway         Interface
-----------        -------         ---------
9.67.1.9/32        0.0.0.0         LIQDIO1
9.67.5.10/32       0.0.0.0         OSAQDIO5L
```

```
NETSTAT ROUTE PR prtable1

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:24:09
Policy Routing Table: prtable1
  IgnorePathMtuUpdate: IPv4: No           IPv6: No
  MultiPath:           IPv4: Conn(Policy) IPv6: Pkt(Profile)
  DynamicXCFRoutes:    IPv4: No           IPv6: No
IPv4 Dynamic Routing Parameters
  Interface        NextHop
  ---------        -------
  OSAQDIOLINK      9.67.115.65
IPv4 Destinations
Destination        Gateway         Flags    Refcnt     Interface
-----------        -------         -----    ------     ---------
Default            9.67.115.65     UGS      0000000002 OSAQDIOLINK
9.67.115.65/32     0.0.0.0         UHS      0000000000 OSAQDIOLINK
9.67.115.69/32     0.0.0.0         UH       0000000000 OSAQDIOLINK
IPv6 Dynamic Routing Parameters
  Interface        NextHop
  ---------        -------
  OSAQDIO46        fe80::9:67:115:65
IPv6 Destinations
DestIP:   Default
  Gw:     2001:0db8::206:2aff:fe71:4400
  Intf:   OSAQDIO46         Refcnt: 0000000000
  Flgs:   UGS               MTU: 1492
DestIP:   2001:0db8::9:67:115:13/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt: 0000000000
  Flgs:   UD                MTU: 1492
DestIP:   2001:0db8::206:2aff:fe71:4400/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt: 0000000000
  Flgs:   UHS               MTU: 1492
```

```
NETSTAT ROUTE RADV

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:40:36
IPv6 Destinations
DestIP:   2001:0db8::206:2aff::/64
  Gw:     fe80::6:2900:6dc:217c
  Intf:   OSAQDIO46  MTU:  1492
```

```
NETSTAT ROUTE RADV DETAIL

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:03:13
IPv6 Destinations
......

DestIP:   2001:0db8::206:2aff::/64
  Gw:     fe80::6:2900:6dc:217c
  Intf:   OSAQDIO46  MTU:  1492
  Metric: 00000001                LifetimeExp: 08/28/2010 15:35
  GwReachable:  Yes               IntfActive:  Yes
  MVS Specific Configured Parameters:
    MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
    RoundTripGain:      0.125     VarianceGain:      0.250
    VarianceMultiplier: 2.000     DelayAcks:         Yes
```

Report field descriptions:

Destination or DestIP
The address of a destination host or network, followed by a slash and the net mask.

Gateway or Gw
The gateway used to send packets to the destination. If the value is 0.0.0.0

for an IPv4 entry or :: for an IPv6 entry, then the destination is directly reachable without needing to go through a gateway.

GwReachable
Indicates whether IPv6 neighbor unreachability detection has detected that the gateway is reachable. The gateway is the router that originates the router advertisement message containing the route.

Yes
Indicates that the gateway is reachable.

No Indicates that the gateway is unreachable. When the gateway is unreachable, the route is not being used for routing. If an alternate route is available, this route is not installed in the routing table.

N/A
The value is displayed for direct prefix router advertisement routes. These routes do not have an associated gateway.

Flags or Flgs
The state of the route, which can have the following values:

G The route uses a gateway.

H The route is to a host rather than to a network.

I The static route in a policy-based routing table is not valid because it is configured to use a link that is not defined in the stack for the same IP version as the route.

U The route is up.

The following flags are mutually exclusive:

C The route was created by a connection (not using a definition or a routing protocol). Routes to subnets or point-to-point destinations using interfaces over which OMPROUTE is active but has not yet established a routing protocol are considered connection routes.

D The route was created dynamically by ICMP processing or router advertisements (IPv6).

O The route was created by OSPF (includes OSPF external routes).

R The route was created by RIP.

S The route is a static route not replaceable by a routing daemon or router advertisements (IPv6).

Z The route is a static route replaceable by dynamic routes learned by OMPROUTE or from router advertisements (IPv6).

Interface or Intf
The link or interface name for the route.

IntfActive
Indicates whether the interface for the route is active.

Yes
Indicates that the interface is active.

No Indicates that the interface is not active. When the interface is not active, the route is not being used for routing. If an alternate route is available, this route is not installed in the routing table.

LifetimeExp

The time at which the IPv6 router advertisement route will be deleted. This lifetime can be extended by router-supplied information. The N/A value is displayed for routes that are advertised with an infinite lifetime.

MTU The value of the maximum transmission unit (MTU). The value can be one of the following depending on the modifier with which the report that contains the MTU value is displayed:

- In a report that is displayed by using neither the RADV modifier nor the RSTAT modifier:
  - If the route is active, this MTU value is the largest packet size that can be sent by using this route. If the packet is larger than the MTU value, the packet must be fragmented if fragmentation is permitted. If fragmentation is not permitted, the packet is dropped and an ICMP error is returned to the originator of the packet.
  - If the route is inactive and was configured by using the BEGINROUTES, RouteTable, or GATEWAY statement, this MTU value is the same as the MTU value that was configured for the route.
  - If the route is inactive and was not configured by using the BEGINROUTES, RouteTable, or GATEWAY statement, this MTU value is either the link MTU value that is received in a router advertisement message for IPv6 routes only or 0.
- In a report that is displayed by using the RADV modifier, this MTU value is the link MTU value that is received in a router advertisement message or 0 if no link MTU value was received.
- In a report that is displayed by using the RSTAT modifier, this MTU value is the MTU value that was configured for the route by using the BEGINROUTES or RouteTable statement.

Metric Displays the metric of the route. For static routes, direct routes have a metric of 0 and indirect routes have a metric of 1. If a route was learned from a routing daemon, the displayed metric is the metric that is set by the routing daemon. If a route was learned from an IPv6 router advertisement, the metric value is one of the following values:

1 When the router advertisement indicates a high preference.

2 When the router advertisement indicates a medium preference.

3 When the router advertisement indicates a low preference.

After the routes are in the stack routing table, the Metric field is not used. The routing daemons use metrics to compare routes and to inform the stack only of the routes that have the best metric.

Maximum retransmit time (MaxReTransmitTime)

The TCP retransmission interval in seconds for this route. If this parameter was not defined for the route, the default value of 120 seconds is displayed. This parameter does not affect initial connection retransmission.

Minimum retransmit time (MinReTransmitTime)

The minimum retransmit interval in seconds for this route. If this parameter was not defined for the route, the default value 0.5 (500 milliseconds) seconds is displayed.

Reference count (RefCnt)

The current number of active users for the route.

Round trip gain (RoundTripGain)

The percentage of the latest round trip time (RTT) to be applied to the smoothed RTT average. The higher this value, the more influence the latest packet RTT has on the average. If this parameter was not defined for the route, the default value 0.125 is displayed. This parameter does not affect initial connection retransmission.

Variance gain (VarianceGain)

The percentage of the latest RTT variance from the RTT average to be applied to the RTT variance average. The higher this value, the more influence the latest packet's RTT has on the variance average. If this parameter was not defined for the route, the default value 0.25 is displayed. This parameter does not affect initial connection retransmission.

Variance multiplier (VarianceMultiplier)

This value is multiplied against the RTT variance in calculating the retransmission interval. [The higher this value, the more effect variation in RTT has on calculating the retransmission interval.] If this parameter was not defined for the route, the default value 2 is displayed. This parameter does not affect initial connection retransmission.

DelayAcks

Indicates whether the DELAYACKS option is enabled or disabled. The value `Yes` indicates that acknowledgments are delayed when a packet is received (the DELAYACKS parameter was defined for the route). The value `No` indicates that acknowledgements are not delayed when a packet is received (the NODELAYACKS parameter was defined for the route).

Policy Routing Table

The name of the policy-based routing table being displayed.

IgnorePathMtuUpdate

IPv4 Indicates whether IPv4 ICMP Fragmentation Needed messages are ignored for this routing table. See the IgnorePathMtuUpdate parameter on the RouteTable statement in the z/OS Communications Server: IP Configuration Reference for more information. This field can have the following values:

Yes IPv4 ICMP Fragmentation Needed messages are ignored for this routing table.

No IPv4 ICMP Fragmentation Needed messages are processed for this routing table.

N/A This value is displayed when either the routing table is not configured for IPv4 routing or only IPv6 routing information is being displayed for the table.

IPv6 Indicates whether IPv6 ICMP Packet Too Big messages are ignored for this routing table. See the IgnorePathMtuUpdate6 parameter on the RouteTable statement in the z/OS Communications Server: IP Configuration Reference for more information. This field can have the following values:

Yes IPv6 ICMP Packet Too Big messages are ignored for this routing table.

No IPv6 ICMP Packet Too Big messages are processed for this routing table.

N/A This value is displayed when either the routing table is not configured for IPv6 routing or only IPv4 routing information is being displayed for the table.

MultiPath

IPv4 The information in this field is divided into two parts. The value before the parentheses indicates whether the multipath routing selection algorithm for outbound IPv4 traffic is enabled for this policy-based routing table. See the multipath parameter on the RouteTable statement in the z/OS Communications Server: IP Configuration Reference for more information. The possible values for the MultiPath field are:

Pkt Indicates that outbound IPv4 traffic uses the round-robin distribution method to use multipath routes for each outbound packet.

Conn Indicates that outbound IPv4 traffic uses the round-robin distribution method to use multipath routes for each outbound connection request.

No Indicates that outbound IPv4 traffic always uses the first active route in a multipath group.

N/A This value is displayed when either the routing table is not configured for IPv4 routing or only IPv6 routing information is being displayed for the table.

The value inside the parentheses identifies where the multipath value was obtained. The possible values are:

Profile
Indicates that the value UseGlobal has been coded on the Multipath parameter on the RouteTable statement; the value was obtained from the IPCONFIG statement.

Policy Indicates that the multipath value was obtained from the Multipath parameter of the RouteTable statement.

Tip: If IPSECURITY is coded on the IPCONFIG statement and Multipath PerPacket is specified on a RouteTable statement, the Multipath PerPacket option is disabled. The value `No(Policy)` is displayed on the report. For more information, see the RouteTable statement information in the z/OS Communications Server: IP Configuration Reference.

IPv6 The information in this field is divided into two parts. The value before the parentheses indicates whether the multipath routing selection algorithm for outbound IPv6 traffic is enabled for this policy-based routing table. See the Multipath6 parameter on the RouteTable statement in the z/OS Communications Server: IP Configuration Reference for more information. The possible values for the MultiPath6 field are:

Pkt Indicates that outbound IPv6 traffic uses the round-robin distribution method to use multipath routes for each outbound packet.

Conn Indicates that outbound IPv6 traffic uses the round-robin distribution method to use multipath routes for each outbound connection request.

No Indicates that outbound IPv6 traffic always uses the first active route in a multipath group.

N/A This value is displayed when either the routing table is not configured for IPv6 routing or only IPv4 routing information is being displayed for the table.

The value inside the parentheses identifies where the multipath value was obtained. The possible values are:

Profile

Indicates that the value UseGlobal has been coded on the Multipath6 parameter on the RouteTable statement; the value was obtained from the IPCONFIG6 statement.

Policy Indicates that the multipath value was obtained from the Multipath6 parameter of the RouteTable statement.

Tip: If IPSECURITY is coded on the IPCONFIG6 statement and Multipath6 PerPacket is specified on a RouteTable statement, the Multipath6 PerPacket option is disabled. The value No(Policy) is displayed on the report. For more information, see the RouteTable statement information in the z/OS Communications Server: IP Configuration Reference.

DynamicXCFRoutes

IPv4 Indicates whether direct routes to the IPv4 dynamic XCF addresses on other TCP/IP stacks are added to the policy-based routing table when the dynamic XCF links to those stacks are active. These are the same routes that are automatically generated in the main routing table when the IPv4 dynamic XCF links are active. See Dynamic XCF in the z/OS Communications Server: IP Configuration Guide for information about the dynamic XCF function and the definitions that are automatically generated when IPCONFIG DYNAMICXCF is specified in the TCP/IP profile. This field can have the following values:

Yes Direct routes to the IPv4 dynamic XCF addresses on other TCP/IP stacks are added to the policy-based routing table when the dynamic XCF links to those stacks are active.

No Direct routes to the IPv4 dynamic XCF addresses on other TCP/IP stacks are not added to the policy-based routing table when the dynamic XCF links to those stacks are active.

N/A This value is displayed when either the routing table is not configured for IPv4 routing or only IPv6 routing information is being displayed for the table.

IPv6 Indicates whether direct routes to the IPv6 dynamic XCF addresses on other TCP/IP stacks are added to the policy-based routing table when the dynamic XCF links to those stacks are active. These are the same routes that are automatically generated in the main routing table when the IPv6 dynamic XCF links are active. See Dynamic XCF in the z/OS Communications Server: IP Configuration Guide for information about the dynamic XCF function and the definitions that are automatically generated when IPCONFIG6 DYNAMICXCF is specified in the TCP/IP profile. This field can have the following values:

Yes Direct routes to the IPv6 dynamic XCF addresses on other TCP/IP stacks are added to the policy-based routing table when the dynamic XCF links to those stacks are active.

No Direct routes to the IPv6 dynamic XCF addresses on other TCP/IP stacks are not added to the policy-based routing table when the dynamic XCF links to those stacks are active.

N/A This value is displayed when either the routing table is not configured for IPv6 routing or only IPv4 routing information is being displayed for the table.

Interface

The name of the interface that is specified in a dynamic routing parameter for the policy-based routing table. If the interface is not currently defined to the TCP/IP stack for the same IP version as the dynamic routing parameter or the interface is inactive on the TCP/IP stack, the name is preceded by an asterisk (*).

NextHop

The next-hop router IP address that is specified in a dynamic routing parameter for the policy-based routing table. The value Any is displayed when no next-hop router IP address is specified for the dynamic routing parameter.

Netstat SLAP/-j report

Displays QoS Policy statistics. By default, all of the QoS policy statistics are displayed. The SUMMARY parameter can be specified to limit the display to summary statistics. Or you can use the POLICYN/**-Y** filter to display only statistics for a specific policy.

TSO syntax:

```
►►──NETSTAT SLAP──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────►◄
```

Modifier:

```
►►──┬──────────────┬──────────────────────────────────────────────►◄
    ├─┤ ACTIVE ├───┤
    └─┤ SUMmary ├──┘
```

ACTIVE
Display QoS policy information only for the activated policies.

SUMmary
Display a summary of QoS policy information.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

Filter:

```
►►──POLicyn──policyname──────────────────────────────────────────►◄
```

z/OS UNIX syntax:

```
►►──netstat -j──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────►◄
```

Modifier:

```
►►──┬──────────────┬──────────────────────────────────────────────►◄
    ├─┤ ACTIVE ├───┤
    └─┤ SUMmary ├──┘
```

ACTIVE
Display QoS policy information only for the activated policies.

SUMmary
Display a summary of QoS policy information.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
►►── -Y──policyname──────────────────────────────────────────────────────►◄
```

Filter description:

POLicyn/-Y *policyname*

Filter the output of the SLAP/**-j** report using the specified policy rule name *policyname*. You can enter one filter value at a time and the specified value can be up to 48 characters long.

The POLicyn/**-Y** filter value can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*".

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single quotation marks. For example, to use an asterisk (*) in the policy name, pgnt*rl for the **-Y** filter, issue the command as: **netstat -j -Y 'pgnt*rl'**.

Command syntax examples:

From TSO environment:

```
NETSTAT SLAP
NETSTAT SLAP SUMMARY
```

From UNIX shell environment:

```
   netstat -j
   netstat -j SUMMARY
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Tip: The Netstat SLAP/**-j** reports are not affected by the IPv6 enablement and format request.

```
NETSTAT SLAP

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:30:49
PolicyRuleName:  ftpd
  FirstActTime:     10/30/2002 20:05:48
  LastMapTime:      10/30/2002 20:06:09
  TotalBytesIn:     34816
  TotalBytesOut:    86016
  TotalInPackets:   17
  TotalOutPackets:  42
  OutBytesInProf:   28672
  OutPacksInProf:   14
  TotalBytesReTrn:  0
  TotalPacksReTrn:  0
  ReTrnTimeouts:    0
  AcceptConn:       5
  DeniedConn:       1
  ActConnMap:       2             Status:            Active
  SmoothRTTAvg:     12            SmoothRTTMdev:     7
  SmoothConnDlyAvg: 5             SmoothConnDlyMdev: 3
  AcceptQDelayAvg:  2             AcceptQDelayMdev:  2
PolicyRuleName:  telnetd
  FirstActTime:     10/30/2002 20:29:53
  LastMapTime:      10/30/2002 20:30:40
  TotalBytesIn:     68
  TotalBytesOut:    108
  TotalInPackets:   2
  TotalOutPackets:  3
  OutBytesInProf:   0
  OutPacksInProf:   0
  TotalBytesReTrn:  0
  TotalPacksReTrn:  0
  ReTrnTimeouts:    0
  AcceptConn:       2
  DeniedConn:       0
  ActConnMap:       2             Status:            Active
  SmoothRTTAvg:     0             SmoothRTTMdev:     0
  SmoothConnDlyAvg: 0             SmoothConnDlyMdev: 0
  AcceptQDelayAvg:  1             AcceptQDelayMdev:  0
```

```
NETSTAT SLAP SUMMARY

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:30:49
PolicyRuleName:  ftpd
  FirstActTime:     10/30/2002 20:05:48
  LastMapTime:      10/30/2002 20:06:09
  Status:           Active
PolicyRuleName:  telnetd
  FirstActTime:     10/30/2002 20:29:53
  LastMapTime:      10/30/2002 20:30:40
  Status:           Active
```

Report field descriptions:

PolicyRuleName
The unique name that identifies the policy rule.

FirstActTime
The time stamp for when the policy rule was first activated.

LastMapTime
The time stamp for when the policy rule was last used.

TotalBytesIn
The number of bytes received by IP for the policy rule.

TotalBytesOut
The number of bytes transmitted by IP for the policy rule.

TotalInPackets
The number of inbound packets received from IP for the policy rule.

TotalOutPackets
The number of outbound packets sent by IP for the policy rule.

OutBytesInProf
This counter counts the number of outbound octets that are determined to be within profile.

OutPacksInProf
This counter counts the number of outbound packets that are determined to be within profile.

TotalBytesReTrn
The number of bytes retransmitted by IP for the policy rule.

TotalPacksReTrn
The number of packets retransmitted by IP for the policy rule.

ReTrnTimeouts
The number of retransmission timeouts for the policy rule.

AcceptConn
This counter is incremented when a policy action (service class) Permission value is set to Allowed and a session (TCP connection) is accepted. It will also be incremented if the policy rule Permission attribute is used.

DeniedConn
This counter is incremented when a policy action Permission value is set to Blocked and a session (TCP connection) is denied, or when a session is rejected due to a policy's connection limit (MaxConnLimit). It will not be incremented if the policy rule Permission attribute is used.

ActConnMap
The number of active TCP connections that are affected by the policy rule.

Status Displays the status of the policy rule. Valid values are Active and Pending Delete. Active indicates that the policy rule is currently in effect. Pending Delete indicates that the policy rule has been marked for deletion but is currently in use. The policy rule is deleted when the rule is no longer in use.

SmoothRTTAvg
The average TCP round trip time for all TCP traffic affected by this policy rule, smoothed over several sampling intervals to reduce large momentary variations.

SmoothRTTMdev
Mean deviation of the TCP round-trip time, smoothed over several sampling intervals to reduce large momentary variations. This value is a computationally less expensive approximation of the standard deviation for this quantity.

SmoothConnDlyAvg
The average connection delay, smoothed over several sampling intervals to reduce large momentary variations. This is the delay between receipt of the first TCP SYN request and the time that the first data packet is returned by the application.

SmoothConnDlyMdev
Mean deviation of the connection delay, smoothed over several sampling

intervals to reduce large momentary variations. This value is a computationally less expensive approximation of the standard deviation for this quantity.

AcceptQDelayAvg
The average accept queue delay. This is the delay between the sending of the TCP SYN ACK for the connection request and the time that the application accepts the connection request.

AcceptQDelayMdev
Mean deviation of the accept queue delay. This value is a computationally less expensive approximation of the standard deviation for this quantity.

Tip: The time displayed in the header of the report is local time. The FirstActTime and LastmapTime fields displayed in the report are Coordinated Universal Time (UTC).

Netstat SOCKets/-s report

Displays information for open TCP and/or UDP sockets associated with a client name.

Restriction: This command displays socket information only for sockets that are bound to a port or an IP address. TCP connections that have been shutdown or aborted are not displayed.

TSO syntax:

```
►►──NETSTAT SOCKets──┤ Target ├──┤ Output ├──┤ (Filter ├────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:

```
                    ┌──────────────────┐
                    ▼                  │
►►──┬─CLIent──────────clientname───────┴──┬────────────────────────────────────────►◄
    ├─HOSTName──hostname──────────────────┤
    │                                     │
    │               ┌─────────────────────┐│
    │               ▼                     ││
    ├─IPAdder─────┬─ipaddr──────────────┬─┘┤
    │             ├─ipaddr/prefixLen────┤  │
    │             └─ipaddr/subnetmask───┘  │
    │                                      │
    │               ┌──────────────────┐   │
    │               ▼                  │   │
    ├─IPPort────────ipaddr+portnum─────┴───┤
    ├─NOTN3270─────────────────────────────┤
    │                                      │
    │            ┌──────────────┐          │
    │            ▼              │          │
    └─POrt──────────portnum─────┴──────────┘
```

z/OS UNIX syntax:

```
►►──netstat -s──┤ Target ├──┤ Output ├──┤ Filter ├─────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
               ┌──────────────────┐
               ▼                  │
►►──┬─ -B──────ipaddr+portnum─────┴──────┬─────────────────────────────────►◄
    │                                    │
    │          ┌──────────────────┐      │
    │          ▼                  │      │
    ├─ -E──────────clientname─────┴──────┤
    ├─ -H──hostname──────────────────────┤
    │                                    │
    │          ┌──────────────────────┐  │
    │          ▼                      │  │
    ├─ -I────┬─ipaddr──────────────┬──┴──┤
    │        ├─ipaddr/prefixLen────┤     │
    │        └─ipaddr/subnetmask───┘     │
    │                                    │
    │          ┌──────────────┐          │
    │          ▼              │          │
    ├─ -P──────────portnum────┴──────────┤
    └─ -T────────────────────────────────┘
```

Filter description:

CLIent/-E *clientname*
: Filter the output of the SOCKets/**-s** report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be up to eight characters long.

HOSTName/-H *hostname*

Filter the output of the SOCKets/**-s** report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters long.

Result: At the end of the report, Netstat will display the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using the HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value, depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the SOCKets/**-s** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength

Filter the output of the SOCKets/**-s** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the SOCKets/**-s** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be an address to which the socket is bound or connected.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and will usually provide the same result as its IPv4 address does.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

IPPort/-B *ipaddr+portnum*

Filter the report output of the SOCKets/-s report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be either the local or remote IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.
- For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.

NOTN3270/-T

Filter the output of the SOCKets/**-s** report, excluding TN3270 server connections.

POrt/-P *portnum*

Filter the output of the SOCKets/**-s** report using the specified port number *portnum*. You can enter up to six filter values.

Guideline: The port number can be a port to which the socket is bound or connected.

Restriction: For a UDP endpoint socket, the filter value applies only to the local or source port.

The filter value for CLIent/**-E** and IPAddr/**-I** can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/**-I** filter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/prefixlen* or *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single (') or double (") quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -s -I '10.*.0.0'** or **netstat -s -I "10.*.0.0"**.

Command syntax examples:

From TSO environment:

```
NETSTAT SOCKETS
   Display information about each client using the socket interface in the default
   TCP/IP stack.
NETSTAT SOCKETS TCP TCPCS6
   Display information about each client using the socket interface in TCPCS6 stack.
NETSTAT SOCKETS TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display information for these clients using the socket interface in TCPCS8 stack
   whose IP addresses to which the socket is bound or connected match the specified
   filter IP address values.
NETSTAT SOCKETS (PORT 2222 6666 88
   Display information for those active TCP connections and UDP sockets in the
   default TCP/IP stack whose port numbers to which the socket is bound or connected
   match the specified filter port numbers.
```

From UNIX shell environment:

```
   netstat -s
   netstat -s -p tcpcs6
   netstat -s -p tcpcs6 -I 9.43.1.1 9.43.2.2
   netstat -s -P 2222 6666 88
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT SOCKETS
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
Sockets interface status:
Type   Bound to               Connected to           State    Conn
====   ========               ============           =====    ====
Name: FTPD1     Subtask: 007E6408
Stream 0.0.0.0..21            0.0.0.0..0             Listen   0000003B
Stream 9.37.65.146..21        9.67.115.5..1026       Establsh 0000003D
Stream 9.37.65.146..21        9.27.13.21..3711       Establsh 0000003F
Name: SYSLOGD1  Subtask: 007E6408
Dgram  0.0.0.0..514           *..*                   UDP      00000010
Name: TAPPV4    Subtask: 007E6460
Dgram  0.0.0.0..2049          9.42.103.99..1234      UDP      00000015

Name: TCPCS     Subtask: 007E2A40
Stream 0.0.0.0..23            0.0.0.0..0             Listen   0000000F
Name: TCPCS     Subtask: 007E08D0
Stream 9.67.115.5..23         9.27.11.182..4886      Establsh 0000000C
```

IPv6 enabled or request for LONG format:

```
NETSTAT SOCKETS
MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS           17:40:36
Sockets interface status:
Name: FTPD1     Subtask: 007E6330
  Type: Stream  Status:  Listen    Conn: 0000004A
    BoundTo: ::..21
    ConnTo:  ::..0
  Type: Stream  Status:  Establsh  Conn: 00000052
    BoundTo: ::ffff:9.67.115.5..21
    ConnTo:  ::ffff:9.67.115.65..1026
  Type: Stream  Status:  Establsh  Conn: 00000058
    BoundTo: 2001:0db8::9:67:115:66..21
    ConnTo:  2001:0db8::9:67:115:65..1027
Name: SYSLOGD1  Subtask: 007E6438
  Type: Dgram   Status:  UDP       Conn: 0000002C
    BoundTo: 0.0.0.0..529
    ConnTo:  *..*
Name: TAPPV4    Subtask: 007E6460
  Type: Dgram   Status:  UDP       Conn: 00000015
    BoundTo: 0.0.0.0..2049
    ConnTo:  9.42.103.99..1234
Name: TAPPV6    Subtask: 007E6480
  Type: Dgram   Status:  UDP       Conn: 00000016
    BoundTo: ::..2050
    ConnTo:  12ab::1..1235

Name: TCPCS     Subtask: 007E1930
  Type: Stream  Status:  Listen    Conn: 0000001A
    BoundTo: 0.0.0.0..23
    ConnTo:  0.0.0.0..0
  Type: Stream  Status:  Establsh  Conn: 0000001E
    BoundTo: 9.67.115.5..23
    ConnTo:  9.27.11.182..4665
Name: USER3     Subtask: 007B93D0
  Type: Stream  Status:  Establsh  Conn: 0000005F
    BoundTo: 2001:0db8::9:67:115:5..1079
    ConnTo:  2001:0db8::9:67:115:65..21
Name: USER6     Subtask: 007B93F0
  Type: Stream  Status:  Establsh  Conn: 000000C7
    BoundTo: 9.67.115.5..1027
    ConnTo:  9.37.65.146..21
```

Report field descriptions:
The following list shows the information displayed after invoking the SOCKets parameter:

Name The client address space name.

Subtask
The subtask identifier indicates the task that created the socket or issued a bind socket API call for the socket. This identifier is the hexadecimal address of the Task Control Block (TCB) associated with this task. The subtask identifier is combined with the address space name to produce a unique identifier for the client.

Type Displays the socket type and can have one of the following values:

Stream
Socket type for stream (TCP) sockets.

Dgram
Socket type for UDP sockets.

Bound to
Indicates the address and port to which the socket is bound. The output is in the format **IP address..bound port** where IP address is the address to

which the socket is bound and bound port is the port number to which the socket is bound. Unbound TCP and UDP sockets are not displayed by NETSTAT CONN.

Connected to
Displays the address and port to which the socket is connected. For UDP sockets, the value of this field is *..* if the socket is not connected. For connected UDP sockets, this field shows the remote IP address and port specified on the connect request. When a UDP socket is connected, it accepts packets only from the specified remote IP address and port.

State Describes the state of the TCP connection. See "TCP connection status" on page 324 for more information.

Conn Displays the client identifier, which is a unique number assigned by the TCP/UDP stack to uniquely identify a socket entity.

Netstat SRCIP/-J report

Displays the job-specific and destination-specific information that is configured using the SRCIP profile statement. See z/OS Communications Server: IP Configuration Reference for more information about the SRCIP statement.

TSO syntax:

```
►►──NETSTAT  SRCIP──┤ Target ├──┤ Output ├──────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

z/OS UNIX syntax:

```
►►──netstat -J──┤ Target ├──┤ Output ├──────────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the **-p** parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT SRCIP
```

From UNIX shell environment:

```
netstat -J
```

Report examples:

The following examples are generated using the TSO NETSTAT command. The z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT SRCIP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:30:49
Source IP Address Based on Job Name:
Job Name  Type  Flg  Source
--------  ----  ---  ------
*         IPV4  C    9.67.5.16
T*        IPV4  S    9.67.5.15
TCPUSR1*  IPV4  B    9.67.5.12
U*        IPV4  C    9.67.5.14
USER1*    IPV4  S    9.67.5.13
USER12    IPV4  B    9.67.5.11

Source IP Address Based on Destination:
Destination         Source
-----------         ------
10.1.0.0/16         9.1.1.2
10.1.1.1            9.1.1.1
```

IPv6 enabled or request for LONG format:

```
NETSTAT SRCIP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:30:49
Source IP Address Based on Job Name:
Job Name  Type  Flg  Source
--------  ----  ---  ------
*         IPV4  C    9.67.5.16
*         IPV6  C    DVIPA66
T*        IPV4  S    9.67.5.15
T*        IPV6  S    2000::9:67:5:15
TCPUSR1*  IPV4  B    9.67.5.12
TCPUSR2*  IPV6  B    DVIPA62
TCPUSR3*  IPV6       TEMPADDRS
TCPUSR4*  IPV6       PUBLICADDRS
U*        IPV4  C    9.67.5.14
U*        IPV6  C    DVIPA64
USER*     IPV6  C    2000::9:67:5:13
USER1*    IPV4  C    9.67.5.13
USER12    IPV4  C    9.67.5.11
U27       IPV6  C    2000::9:67:5:11

Source IP Address Based on Destination:
Destination: 10.1.0.0/16
  Source:    9.1.1.2
Destination: 10.1.1.1
  Source:    9.1.1.1
Destination: 2001:0db8::0522:f103
  Source:    2000::9:67:5:10
Destination: 2001:0db8::/32
  Source:    DVIPA66
```

Report field descriptions:

Destination

A destination IP address, network address, or subnet address for which the designated source should be used to provide the source IP address for an outbound TCP connection. If a connection's destination address matches more than one Destination value, the most complete match is selected. The Destination designations are ignored if a connection's job name matches a Job Name value with at least one non-wildcard character, but a Destination match overrides a JOBNAME * match.

Job Name
The name of the job that matches this entry. The job name can end in an asterisk (*). Any job that is running that begins with the same characters that precede the asterisk matches this designation. If several different designations exist, then the matching entry is determined by the most complete match: either an exact match, or the entry that has the most matching characters before the asterisk in the job-specific source IP address designation. An asterisk (*) in this field indicates that all of the applications match the entry.

When an IP address or interface name is displayed in the Source column, the Job Name value is the name of the job or jobs for which the designated interface should be used as the source IP address. An asterisk (*) indicates that all applications that issue TCP connect requests are associated with the specified source IP address or interface. They override any existing TCPSTACKSOURCEVIPA specifications except for outbound connections whose destination address matches a Destination value.

When `TEMPADDRS` is displayed in the Source column, the Job Name value is the name of the job or jobs that should prefer a temporary IPv6 address over a public IPv6 address when the default source address selection algorithm is used to select the source IP address. When PUBLICADDRS is displayed in the Source column, the Job Name value is the name of the job or jobs that should prefer a public IPv6 address over a temporary IPv6 address when the default source address selection algorithm is used to select the source IP address. An asterisk (*) in the Job Name column indicates that all the applications match the entry. See the information about default source address selection in z/OS Communications Server: IPv6 Network and Application Design Guide.

Type
The address family to which this job-specific source IP address applies, either IPv4 or IPv6. An entry for which the value `TEMPADDRS` or `PUBLICADDRS` is displayed in the Source column always has the value `IPv6`.

Flg
The flags represent parameter values defined with the JOBNAME parameter on the SRCIP profile statement.

B
The value Both was specified for this SRCIP JOBNAME statement. This JOBNAME statement is used for both TCP client and server applications. For server applications it is applied for only servers that invoke the bind() function call with the IPv4 INADDR_ANY address or the IPv6 unspecified address (in6addr_any).

C
The value Client was specified for this SRCIP JOBNAME statement (or was set by default). This JOBNAME statement is used for TCP outbound (client) connections only.

S
The value Server was specified for this SRCIP JOBNAME statement. This JOBNAME statement is used for server applications that invoke the bind() function call with the IPv4 INADDR_ANY address or the IPv6 unspecified address (in6addr_any).

This field is blank for an entry that displays the value TEMPADDRS or PUBLICADDRS in the Source column.

Source

IP address or interface name
The interface name or IP address that is used to supply a source IP address for TCP client or server applications.

- When the source address is displayed after the destination display line, TCP client applications that have a destination IP address that matches the corresponding destination value use this source address.
- When the source address is displayed with job name values, both IPv4 and IPv6 TCP client and server applications that have a job name that matches the corresponding job name value use this source address depending on the flag field value (Both, Client only, or Server only).

TEMPADDRS
Indicates that a temporary IPv6 address should be preferred over a public IPv6 address when IPv6 default address selection is used to select the source IP address for the specified job.

PUBLICADDRS
Indicates that a public IPv6 address should be preferred over a temporary IPv6 address when IPv6 default address selection is used to select the source IP address for the specified job.

Netstat STATS/-S report

Displays TCP/IP statistics for IP, ICMP, TCP, and UDP protocols. You can use the PROTOCOL filter to display statistics for only a specific protocol.

TSO syntax:

```
►►──NETSTAT STATS──┤ Modifier ├──┤ Target ├──┤ Output ├──────────────────────►◄
```

Modifier:

```
►►──┬──────────┬──┬──────────┬──────────────────────────────────────────────►◄
    └─PROTOcol─┘  └─protocol─┘
```

PROTOcol *protocol*
Display statistics for the specified protocol. The valid protocols are IP, ICMP, TCP, and UDP.

Result: If you specify TCP, you get both TCP and SMC-R statistics.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

z/OS UNIX syntax:

```
►►─netstat -S──┤ Modifier ├──┤ Target ├──┤ Output ├─────────────────────────►◄
```

Modifier:

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─PROTOcol─┘  └─protocol─┘
```

PROTOcol *protocol*

Display statistics for the specified protocol. The valid protocols are IP, ICMP, TCP, and UDP.

Result: If you specify TCP, you get both TCP and SMC-R statistics. For more information about SMC-R support, see Shared Memory Communications over Remote Direct Memory Access in the z/OS Communications Server: IP Configuration Guide.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT STATS
   Provides TCP/IP statistics for IP, ICMP, TCP and UDP protocols.
NETSTAT STATS PROTOCOL IP
   Provides TCP/IP statistics for IP protocol. If the stack is IPv6-enabled, then the
   statistics for IPv6 protocol are also displayed.
NETSTAT STATS PROTOCOL ICMP
   Provides TCP/IP statistics for ICMP protocol. If the stack is IPv6-enabled, then
   the statistics for ICMPv6 protocol are also displayed.
NETSTAT STATS PROTOCOL TCP
   Provides TCP/IP statistics for TCP protocol. If the stack is enabled for SMC-R,
   then the statistics for SMC-R are also displayed.
NETSTAT STATS PROTOCOL UDP
   Provides TCP/IP statistics for UDP protocol.
```

From UNIX shell environment:

```
   netstat -S
   netstat -S PROTOCOL IP
   netstat -S PROTOCOL ICMP
   netstat -S PROTOCOL TCP
   netstat -S PROTOCOL UDP
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT STATS

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:14:15
IP Statistics
  Packets Received                   = 25164
  Inbound Calls from Device Layer    = 12241
  Inbound Frame Unpacking Errors     = 0
  Inbound Discards Memory Shortage   = 0
  Received Header Errors             = 0
  Received Address Errors            = 4961
  Datagrams Forwarded                = 067
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 3
  Received Packets Delivered         = 20203
  Output Requests                    = 8773
  Output Discards No Route           = 0
  Output Discards DLC Sync Errors    = 0
  Output Discards DLC Async Errors   = 0
  Output Discards Memory Shortage    = 0
  Output Discards (other)            = 0
  Reassembly Timeouts                = 0
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0
  Inbound  Packets handled by zIIP   = 12490
  Outbound Packets handled by zIIP   = 4912
```

```
ICMP Statistics
                              Received    Sent
                              --------    ----
  Messages                    1366        7
  Errors                      0           0
  Destination Unreachable     1359        0
  Time Exceeded               0           0
  Parameter Problems          0           0
  Source Quenchs              0           0
  Redirects                   0           0
  Echos                       7           0
  Echo Replies                0           7
  Timestamps                  0           0
  Timestamp Replies           0           0
  Address Masks               0           0
  Address Mask Replies        0           0

TCP Statistics
  Current Established Connections     = 11
  Current Stalled Connections         = 0
  Current Servers In Connection Flood = 0
  Active Connections Opened           = 122
  Passive Connections Opened          = 7
  Connections Closed                  = 78
  Established Connections Dropped     = 8
  Connection Attempts Dropped         = 4
  Connection Attempts Discarded       = 2
  Timewait Connections Reused         = 0
  Segments Received                   = 10900
  Header Prediction Ok for ACK        = 1643
  Header Prediction Ok for Data       = 3213
  Duplicate ACKs                      = 134
  Discards for Bad Checksum           = 0
  Discards for Bad Header Length      = 0
  Discards for Data too Short         = 9
  Discards for Old Timestamp          = 2
  Segments Completely Duplicate       = 23
  Segments Partially Duplicate        = 4
  Segments Completely After Window    = 0
  Segments Partially After Window     = 0
  Segments Out of Order               = 43
  Segments Received After Close       = 2
  Window Probes Received              = 5
  Window Updates Received             = 9
  Segments Received on OSA Bulk Queues= 9
  Segments Sent                       = 8382
  Window Updates Sent                 = 723
  Delayed ACKs Sent                   = 43
  Resets Sent                         = 4
  Segments Retransmitted              = 21
  Retransmit Timeouts                 = 0
  Connections Dropped by Retransmit   = 0
  Path MTU Discovery Retransmits      = 0
  Path MTU Beyond Retransmit Limit    = 0
  Window Probes Sent                  = 2
  Connections Dropped during Probe    = 0
  KeepAlive Probes Sent               = 0
  Connections Dropped by KeepAlive    = 0
  Connections Dropped by Finwait2     = 0
  Configured Ephemeral Ports          = 200
  Configured Ephemeral Ports In Use   = 5
  Configured Ephemeral Ports Max Usage= 5
  Ephemeral Ports Exhausted           = 0
```

```
SMCR Statistics
  Current Established SMC Links       = 2
  SMC Link Activation Time Outs       = 0
  Active SMC Links Opened             = 4
  Passive SMC Links Opened            = 0
  SMC Links Closed                    = 2
  Current Established Connections     = 1
  Active Connections Opened           = 1
  Passive Connections Opened          = 0
  Connections Closed                  = 0
  Segments Received                   = 1
  Segments Sent                       = 1
  Resets Sent                         = 0
  Resets Received                     = 0
UDP Statistics
  Datagrams Received    = 6984
  No Port Errors        = 2312
  Receive Errors        = 0
  Datagrams Sent        = 368
  Configured Ephemeral Ports          = 200
  Configured Ephemeral Ports In Use   = 6
  Configured Ephemeral Ports Max Usage= 7
  Ephemeral Ports Exhausted           = 0
```

```
NETSTAT STATS PROTOCOL IP

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:14:15
IP Statistics
  Packets Received                   = 25164
  Inbound Calls from Device Layer    = 12241
  Inbound Frame Unpacking Errors     = 0
  Inbound Discards Memory Shortage   = 0
  Received Header Errors             = 0
  Received Address Errors            = 4961
  Datagrams Forwarded                = 067
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 3
  Received Packets Delivered         = 20203
  Output Requests                    = 8773
  Output Discards No Route           = 0
  Output Discards DLC Sync Errors    = 0
  Output Discards DLC Async Errors   = 0
  Output Discards Memory Shortage    = 0
  Output Discards (other)            = 0
  Reassembly Timeouts                = 0
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0
  Inbound  Packets handled by zIIP   = 12490
  Outbound Packets handled by zIIP   = 4912
```

```
NETSTAT STATS PROTOCOL ICMP

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:14:15
ICMP Statistics
                              Received    Sent
                              --------    ----
  Messages                    1366        7
  Errors                      0           0
  Destination Unreachable     1359        0
  Time Exceeded               0           0
  Parameter Problems          0           0
  Source Quenchs              0           0
  Redirects                   0           0
  Echos                       7           0
  Echo Replies                0           7
  Timestamps                  0           0
  Timestamp Replies           0           0
  Address Masks               0           0
  Address Mask Replies        0           0
```

IPv6 enabled or request for LONG format:

```
NETSTAT STATS

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:14:15
IP Statistics (IPv4)
  Packets Received                  = 34
  Received Header Errors            = 0
  Received Address Errors           = 3
  Datagrams Forwarded               = 0
  Unknown Protocols Received        = 0
  Received Packets Discarded        = 3
  Received Packets Delivered        = 46
  Output Requests                   = 31
  Output Discards No Route          = 0
  Output Discards (other)           = 0
  Reassembly Timeouts               = 0
  Reassembly Required               = 0
  Reassembly Successful             = 0
  Reassembly Failures               = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation   = 0
  Fragments Created                 = 0
  Inbound  Packets handled by zIIP   = 12490
  Outbound Packets handled by zIIP   = 4912
IPv6 Statistics
  Packets Received                  = 0
  Received Header Errors            = 0
  Received Address Errors           = 0
  Datagrams Forwarded               = 0
  Unknown Protocols Received        = 0
  Received Packets Discarded        = 0
  Received Packets Delivered        = 0
  Output Requests                   = 0
  Output Discards No Route          = 0
  Output Discards (other)           = 0
  Reassembly Timeouts               = 0
  Reassembly Required               = 0
  Reassembly Successful             = 0
  Reassembly Failures               = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation   = 0
  Fragments Created                 = 0
  Inbound  Packets handled by zIIP   = 0
  Outbound Packets handled by zIIP   = 0
IP General Statistics
  Inbound Calls from Device Layer   = 91
  Inbound Frame Unpacking Errors    = 0
  Inbound Discards Memory Shortage   = 0
  Output Discards DLC Sync Errors   = 0
  Output Discards DLC Async Errors   = 0
  Output Discards Memory Shortage   = 0
```

```
ICMP Statistics (IPV4)
                              Received    Sent
                              --------    ----
  Messages                    12          12
  Errors                      0           12
  Destination Unreachable     12          12
  Time Exceeded               0           0
  Parameter Problems          0           0
  Source Quenchs              0           0
  Redirects                   0           0
  Echos                       0           0
  Echo Replies                0           0
  Timestamps                  0           0
  Timestamp Replies           0           0
  Address Masks               0           0
  Address Mask Replies        0           0
ICMPv6 Statistics
                              Received    Sent
                              --------    ----
  Messages                    0           4
  Errors                      0           0
  Destination Unreachable     0           0
  Time Exceeded               0           0
  Parameter Problems          0           0
  Redirects                   0           0
  Echos                       0           0
  Echo Replies                0           0
  Administratively Prohibited 0           0
  Packet Too Big              0           0
  Router Solicitations        0           0
  Router Advertisements       0           0
  Neighbor Solicitations      0           0
  Neighbor Advertisements     0           0
  Group Membership Queries    0           0
  Group Membership Responses  0           4
  Group Membership Reductions 0           0
```

```
TCP Statistics
  Current Established Connections     = 2
  Current Stalled Connections         = 0
  Current Servers In Connection Flood = 0
  Active Connections Opened           = 1
  Passive Connections Opened          = 1
  Connections Closed                  = 0
  Established Connections Dropped     = 0
  Connection Attempts Dropped         = 0
  Connection Attempts Discarded       = 0
  Timewait Connections Reused         = 0
  Segments Received                   = 6
  Header Prediction Ok for ACK        = 0
  Header Prediction Ok for Data       = 2
  Duplicate ACKs                      = 0
  Discards for Bad Checksum           = 0
  Discards for Bad Header Length      = 0
  Discards for Data too Short         = 0
  Discards for Old Timestamp          = 0
  Segments Completely Duplicate       = 0
  Segments Partially Duplicate        = 0
  Segments Completely After Window    = 0
  Segments Partially After Window     = 0
  Segments Out of Order               = 0
  Segments Received After Close       = 0
  Window Probes Received              = 0
  Window Updates Received             = 0
  Segments Received on OSA Bulk Queues= 9
  Segments Sent                       = 7
  Window Updates Sent                 = 0
  Delayed ACKs Sent                   = 2
  Resets Sent                         = 0
  Segments Retransmitted              = 0
  Retransmit Timeouts                 = 0
  Connections Dropped by Retransmit   = 0
  Path MTU Discovery Retransmits      = 0
  Path MTU Beyond Retransmit Limit    = 0
  Window Probes Sent                  = 0
  Connections Dropped during Probe    = 0
  KeepAlive Probes Sent               = 0
  Connections Dropped by KeepAlive    = 0
  Connections Dropped by Finwait2     = 0
  Configured Ephemeral Ports          = 200
  Configured Ephemeral Ports In Use   = 5
  Configured Ephemeral Ports Max Usage= 5
  Ephemeral Ports Exhausted           = 0
SMCR Statistics
  Current Established SMC Links       = 2
  SMC Link Activation Time Outs       = 0
  Active SMC Links Opened             = 4
  Passive SMC Links Opened            = 0
  SMC Links Closed                    = 2
  Current Established Connections     = 1
  Active Connections Opened           = 1
  Passive Connections Opened          = 0
  Connections Closed                  = 0
  Segments Received                   = 1
  Segments Sent                       = 1
  Resets Sent                         = 0
  Resets Received                     = 0
UDP Statistics
  Datagrams Received    = 0
  No Port Errors        = 12
  Receive Errors        = 0
  Datagrams Sent        = 12
  Configured Ephemeral Ports          = 200
  Configured Ephemeral Ports In Use   = 6
  Configured Ephemeral Ports Max Usage= 7
  Ephemeral Ports Exhausted           = 0
```

```
NETSTAT STATS PROTOCOL IP

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:14:15
IP Statistics (IPv4)
  Packets Received                   = 34
  Received Header Errors             = 0
  Received Address Errors            = 3
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 3
  Received Packets Delivered         = 46
  Output Requests                    = 31
  Output Discards No Route           = 0
  Output Discards (other)            = 0
  Reassembly Timeouts                = 0
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0
  Inbound  Packets handled by zIIP   = 12490
  Outbound Packets handled by zIIP   = 4912
IPv6 Statistics
  Packets Received                   = 0
  Received Header Errors             = 0
  Received Address Errors            = 0
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 0
  Received Packets Delivered         = 0
  Output Requests                    = 0
  Output Discards No Route           = 0
  Output Discards (other)            = 0
  Reassembly Timeouts                = 0
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0
  Inbound  Packets handled by zIIP   = 0
  Outbound Packets handled by zIIP   = 0
IP General Statistics
  Inbound Calls from Device Layer    = 91
  Inbound Frame Unpacking Errors     = 0
  Inbound Discards Memory Shortage   = 0
  Output Discards DLC Sync Errors    = 0
  Output Discards DLC Async Errors   = 0
  Output Discards Memory Shortage    = 0
```

```
NETSTAT STATS PROTOCOL ICMP

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          15:14:15
ICMP Statistics (IPV4)
                              Received    Sent
                              --------    ----
  Messages                    12          12
  Errors                      0           12
  Destination Unreachable     12          12
  Time Exceeded               0           0
  Parameter Problems          0           0
  Source Quenchs              0           0
  Redirects                   0           0
  Echos                       0           0
  Echo Replies                0           0
  Timestamps                  0           0
  Timestamp Replies           0           0
  Address Masks               0           0
  Address Mask Replies        0           0
ICMPv6 Statistics
                              Received    Sent
                              --------    ----
  Messages                    0           4
  Errors                      0           0
  Destination Unreachable     0           0
  Time Exceeded               0           0
  Parameter Problems          0           0
  Redirects                   0           0
  Echos                       0           0
  Echo Replies                0           0
  Administratively Prohibited 0           0
  Packet Too Big              0           0
  Router Solicitations        0           0
  Router Advertisements       0           0
  Neighbor Solicitations      0           0
  Neighbor Advertisements     0           0
  Group Membership Queries    0           0
  Group Membership Responses  0           4
  Group Membership Reductions 0           0
```

Report field descriptions:

Most of the TCP/IP statistics for IP, ICMP, TCP, and UDP protocols are defined in the SNMP IP-MIB (RFC2011 - *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2*), TCP-MIB (RFC 2012 - *SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2*), and UDP-MIB (RFC 2013 - *SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2*) MIB modules. See these SNMP MIB modules for more detailed information.

- The following describes the IPv4 and IPv6 statistics displayed:

  Packets Received
  The total number of input datagrams received from interfaces.

  Received Header Errors
  The number of input datagrams discarded due to errors in their IP headers.

  Received Address Errors
  The number of input datagrams discarded because the IP address in their IP header's destination field was not valid.

  Datagrams Forwarded
  The number of input datagrams forwarded to their final destination.

  Unknown Protocols Received
  The number of datagrams discarded because of an unknown or unsupported protocol.

Received Packets Discarded
The number of input datagrams that were discarded that are not accounted for in another input discard counter.

Received Packets Delivered
The total number of input datagrams successfully delivered to IP user-protocols.

Output Requests
The total number of IP datagrams that local IP user-protocols supplied to IP in requests for transmission.

Output Discards No Route
The number of IP datagrams discarded because no route could be found to transmit them to their destination.

Output Discards (Other)
The number of output datagrams generated by this stack that could not be transmitted.

Reassembly Timeouts
The number of packets that were being held for reassembly but which were discarded due to the fact that the remaining fragments were not received within reassembly timeout.

Reassembly Required
The number of IP fragments received that needed to be reassembled.

Reassembly Successful
The number of IP datagrams successfully reassembled.

Reassembly Failures
The number of failures detected by the IP reassembly algorithm.

Datagrams Successfully Fragmented
The number of IP datagrams that have been successfully fragmented.

Datagrams Failing Fragmentation
The number of IP datagrams that have been discarded because they needed to be fragmented but could not be.

Fragments Created
The number of IP datagram fragments that have been generated as a result of fragmentation.

Inbound Packets handled by zIIP
The number of inbound packets that were processed by a zIIP. This counter applies only to IPSec workloads, whose CPU cycles are being displaced to a zIIP. The Packets Received counter includes the packets that are received on zIIP, so the percentage of total inbound packets that were processed by zIIP can be calculated as (Inbound Packets handled by zIIP ÷ Packets Received) × 100. Similarly, the number of inbound packets that were processed by General Purpose Processors is equal to (Packets Received - Inbound Packets handled by zIIP).

Outbound Packets handled by zIIP
The number of outbound packets that were processed by a zIIP. This counter applies only to IPSec workloads, whose CPU cycles are being displaced to a zIIP. The Output Requests counter includes the outbound packets processed on zIIP, so the percentage of total outbound packets that were processed by zIIP can be calculated as (Outbound Packets handled by zIIP ÷ Output Requests) × 100. Similarly, the number of

outbound packets that were processed by General Purpose Processors is equal to (Output Requests - Outbound Packets handled by zIIP).

- The following describes the IP general statistics displayed. The statistic values for these counters reflect both IPv4 and IPv6 processing combined.

  Inbound Calls from Device Layer
  The number of times the inbound TCP/IP Data Path has received control from the Device Layer.

  Inbound Frame Unpacking Errors
  The number of times a received frame could not be unpacked into its constituent datagrams.

  Inbound Discards Memory Shortage
  The number of inbound packets discarded due to a CSM storage shortage condition.

  Output Discards DLC Sync Errors
  The number of outbound packets discarded due to a synchronous error in the Data Link Control.

  Output Discards DLC Async Errors
  The number of outbound packets discarded due to an asynchronous error in the Data Link Control.

  Output Discards Memory Shortage
  The number of outbound packets discarded due to a CSM storage shortage condition.

- The following describes the ICMP statistics displayed:

  Messages
  The total number of ICMP messages received and sent.

  Errors The number of ICMP messages received and sent but determined as having ICMP-specific errors.

  Destination Unreachable
  The number of ICMP Destination Unreachable messages received and sent.

  Time Exceeded
  The number of ICMP Time Exceeded messages received and sent.

  Parameter Problems
  The number of ICMP Parameter Problem messages received and sent.

  Source Quenchs
  The number of ICMP Source Quench messages received and sent.

  Redirects
  The number of ICMP Redirect messages received and sent.

  Echos The number of ICMP Echo (request) messages received and sent.

  Echo Replies
  The number of ICMP Echo Reply messages received and sent.

  Timestamps
  The number of ICMP Timestamp (request) messages received and sent.

  Timestamp Replies
  The number of ICMP Timestamp Reply messages received and sent.

Address Masks
The number of ICMP Address Mask (request) messages received and sent.

Address Mask Replies
The number of ICMP Address Mask Reply messages received and sent.

- The following describes the ICMPv6 statistics displayed:

Messages
The total number of ICMPv6 messages received and sent.

Errors
The number of ICMPv6 messages received and sent but determined as having ICMPv6-specific errors.

Destination Unreachable
The number of ICMPv6 Destination Unreachable messages received and sent.

Time Exceeded
The number of ICMPv6 Time Exceeded messages received and sent.

Parameter Problems
The number of ICMPv6 Parameter Problem messages received and sent.

Redirects
The number of ICMPv6 Redirect messages received and sent.

Echos
The number of ICMPv6 Echo messages received and sent.

Echo Replies
The number of ICMPv6 Echo Reply messages received and sent.

Administratively Prohibited
The number of ICMPv6 Administratively Prohibited messages received and sent.

Packet Too Big
The number of ICMPv6 Packet Too Big messages received and sent.

Router Solicitations
The number of ICMPv6 Router Solicitation messages received and sent.

Router Advertisements
The number of ICMPv6 Router Advertisement messages received and sent.

Neighbor Solicitations
The number of ICMPv6 Neighbor Solicitation messages received and sent.

Neighbor Advertisements
The number of ICMPv6 Neighbor Advertisement messages received and sent.

Group Membership Queries
The number of ICMPv6 Group Membership Queries received and sent.

Group Membership Responses
The number of ICMPv6 Group Membership Responses received and sent.

Group Membership Reductions
The number of ICMPv6 Group Membership Reductions received and sent.

- The following describes the TCP statistics displayed:

  Current Established Connections

  The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.

  Guideline: This value, when displayed for TCP statistics, includes the number of TCP connections across SMC-R links. To determine the number of TCP connections that are not across SMC-R links, subtract the value for `Current Established Connections` displayed under `SMCR Statistics` from this value.

  Current Stalled Connections

  The number of TCP connections whose send data flow is stalled. The send data flow is considered stalled if one or more of the following conditions are true:

  - The TCP send window size is less than 256 or is less than the smaller of the largest send window that has been seen for the connection and the default MTU. The TCP send window size is set based on values provided by the TCP peer. The default MTU for IPv4 is 576. The default MTU for IPv6 is 1280.
  - The TCP send queue is full and the data is not being retransmitted.

  Current Servers In Connection Flood

  The number of TCP servers under a potential connection flood attack. A server is considered under a potential connection flood attack when backlog queue expansion is required to handle the incoming connection requests. When more than 25 servers are under a potential connection flood attack, no server's backlog queue will be allowed to expand.

  Active Connections Opened

  The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

  Guideline: This value, when displayed for TCP statistics, includes the number of TCP connections across SMC-R links that made a direct transition to the SYN-SENT state from the CLOSED state. To determine the number of these TCP connections that are not across SMC-R links, subtract the value for `Active Connections Opened` displayed under `SMCR Statistics` from this value.

  Passive Connections Opened

  The number of times that TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state. This value, when displayed for TCP statistics, includes the number of TCP connections across SMC-R links that made a direct transition to the SYN-RCVD state from the LISTEN state. To determine the number of the TCP connections that are not across SMC-R links, subtract the value for `Passive Connections Opened` displayed under `SMCR Statistics` from this value.

  Connections Closed

  Number of TCP connections that have corresponding sockets closed.

  Guideline: This value, when displayed for TCP statistics, includes the number of TCP connections across SMC-R links that have corresponding sockets closed. To determine the number of these TCP connections that are not across SMC-R links, subtract the value for `Connections Closed` displayed under `SMCR Statistics` from this value.

Established Connections Dropped
The number of times TCP connections made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state. This value includes the number of TCP connections across SMC-R links.

Connection Attempts Dropped
The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the listen state from the SYN-RCVD state.

Connection Attempts Discarded
Number of passive connection requests discarded.

Timewait Connections Reused
Number of TCP connections in the TIMEWAIT state that have been reused for connections in the SYN-RCVD state.

Segments Received
The total number of segments received.

Guideline: This value, when displayed for TCP statistics, includes the number of Remote Direct Memory Access (RDMA) inbound operations that are processed across all SMC-R links. To determine the number of segments that are received on TCP connections that do not traverse SMC-R links, subtract the value for `Segments Received` displayed under `SMCR Statistics` from this value.

Segments Received on OSA Bulk Queues
The total number of segments received for all connections from the BulkData ancillary input queue (AIQ) of the OSA-Express QDIO inbound workload queueing function. For more information about QDIO inbound workload queueing, see z/OS Communications Server: IP Configuration Guide.

Header Prediction Ok for ACK
Number of inbound TCP acknowledgments with successful header prediction.

Header Prediction Ok for Data
Number of inbound TCP data segments with successful header prediction.

Duplicate ACKs
Number of inbound duplicate TCP acknowledgments.

Discards for Bad Checksum
Number of inbound TCP segments discarded due to bad checksum.

Discards for Bad Header Length
Number of inbound TCP segments discarded due to bad header length.

Discards for Data too Short
Number of inbound TCP segments discarded due to data length shorter than segment length.

Discards for Old Timestamp
Number of inbound TCP segments discarded due to old timestamp.

Segments Completely Duplicate
Number of inbound TCP segments with all data before current TCP window.

Segments Partially Duplicate
Number of inbound TCP segments with some data before current TCP window.

Segments Completely After Window
Number of inbound TCP segments with all data after current TCP window.

Segments Partially After Window
Number of inbound TCP segments with some data after current TCP window.

Segments Out of Order
Number of inbound TCP segments that did not contain the next expected sequence number.

Segments Received After Close
Number of inbound TCP segments received after corresponding sockets have been closed.

Window Probes Received
Number of inbound TCP segments processed while current receive window size is 0.

Window Updates Received
Number of inbound TCP segments that only change receive window size.

Segments Sent
The total number of segments sent.

Guideline: This value, when displayed for TCP statistics, includes the number of RDMA outbound operations that are processed across all SMC-R links. To determine the number of segments that were sent on TCP connections that do not traverse SMC-R links, subtract the value for `Segments Sent` displayed under `SMCR Statistics` from this value.

Window Updates Sent
Number of outbound TCP segments that only change receive window size.

Delayed ACKs Sent
Number of delayed outbound TCP acknowledgments.

Resets Sent
Number of TCP segments sent containing the RST flag.

Guideline: This value, when displayed for TCP statistics, includes the number of TCP connections that were using SMC-R links. To determine the number of these segments that were sent for TCP connections that were not using SMC-R links, subtract the value for `Resets Sent` displayed under `SMCR Statistics` from this value.

Segments Retransmitted
The total number of segments retransmitted.

Retransmit Timeouts
Number of TCP retransmit timer pops.

Connections Dropped by Retransmit
Number of TCP connections dropped due to retransmit threshold exceeded.

Path MTU Discovery Retransmits
Number of outbound TCP segments retransmitted due to path MTU discovery.

Path MTU Beyond Retransmit Limit
Number of TCP connections that exceeded path MTU discovery retransmit threshold.

Window Probes Sent
Number of outbound window probe requests.

Connections Dropped during Probe
Number of TCP connections dropped due to no response while sending window probe requests.

KeepAlive Probes Sent
Number of keepalive probe requests. This value includes the number of TCP connections across SMC-R links.

Connections Dropped by KeepAlive
Number of TCP connections dropped because of no response when sending keepalive probe requests. This value includes the number of TCP connections across SMC-R links.

Connections Dropped by Finwait2
Number of TCP connections dropped because of FINWAIT2 timer expiring before receiving FIN segments. This value includes the number of TCP connections across SMC-R links.

Configured Ephemeral Ports
Number of configured ephemeral ports to be assigned for TCP applications.

Ephemeral Ports In Use
The number of ephemeral ports currently in use by TCP applications.

Ephemeral Ports Max Usage
The highest number of ephemeral ports in use by TCP applications at any time.

Ephemeral Ports Exhausted
The number of times a bind() request failed because all available ephemeral ports were in use.

- The following describes the SMC-R statistics that are displayed:

  Current Established SMC Links
  The current number of active SMC-R links.

  SMC Link Activation Time Outs
  The number of times that an attempt occurred to establish an SMC-R link, but the attempt failed because of timeout conditions.

  Active SMC Links Opened
  The number of times that an SMC-R link was established and this stack acted in the server role during link establishment.

  Passive SMC Links Opened
  The number of times that an SMC-R link was established and this stack acted in the client role during link establishment.

SMC Links Closed
The number of SMC-R links that were closed.

Current Established Connections
The number of TCP connections over SMC-R links for which the current state is either ESTABLISHED or CLOSE-WAIT.

Active Connection Opened
The number of times that TCP connections over SMC-R links made a direct transition to the SYN-SENT state from the CLOSED state.

Passive Connection Opened
The number of times that TCP connections over SMC-R links made a direct transition to the SYN-RCVD state from the LISTEN state.

Connections Closed
The number of TCP connections over SMC-R links that have corresponding sockets closed.

Segments Received
The number of Remote Direct Memory Access (RDMA) inbound operations that were processed across all SMC-R links.

Segments Sent
The number of RDMA outbound operations that were processed across all SMC-R links.

Resets Sent
The number of RDMA outbound operations that contained the abnormal close flag. For information about the abnormal close flag, see z/OS Communications Server: IP Configuration Guide.

Resets Received
The number of RDMA inbound operations that contained the abnormal close flag. For information about the abnormal close flag, see z/OS Communications Server: IP Configuration Guide.

- The following describes the UDP statistics displayed:

Datagrams Received
The total number of UDP datagrams delivered to UDP users.

No Port Errors
The total number of received UDP datagrams for which there was no application at the destination port.

Receive Errors
The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Datagrams Sent
The total number of UDP datagrams sent.

Configured Ephemeral Ports
Number of configured ephemeral ports to be assigned for UDP applications.

Ephemeral Ports In Use
The number of ephemeral ports currently in use by UDP applications.

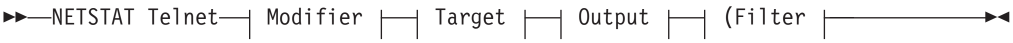
Ephemeral Ports Max Usage
The highest number of ephemeral ports in use by UDP applications at any time.

Ephemeral Ports Exhausted
The number of times a bind() request failed because all available ephemeral ports were in use.

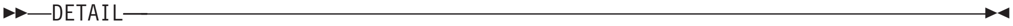
Netstat TELnet/-t report

Displays information for TN3270E Telnet server connections.

TSO syntax:



Modifier:

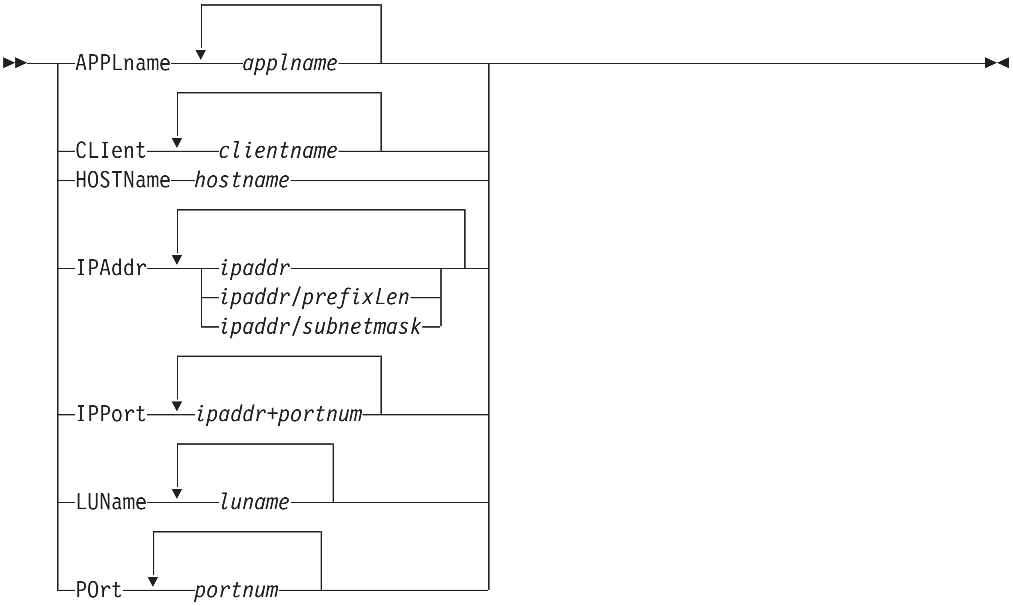


DETAIL
Displays the logmode and Telnet protocol in use by each connection. If an application user ID was entered on the solicitor panel, it is displayed in the TnUserId field. Otherwise, the TnUserId field is blank.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:



z/OS UNIX syntax:

```
►►──netstat -t──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────────►◄
```

Modifier:

```
►►──DETAIL──────────────────────────────────────────────────────────────────────►◄
```

DETAIL
Displays the logmode and Telnet protocol in use by each connection. If an application user ID was entered on the solicitor panel, it is displayed in the TnUserId field. Otherwise, the TnUserId field is blank.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:

```
              ┌───────────────┐
►►──┬─ -B──▼──ipaddr+portnum──┴────────┬──────────────────────────────►◄
    │         ┌───────────────┐        │
    ├─ -E──▼────clientname────┴────────┤
    ├─ -H──hostname────────────────────┤
    │         ┌──────────────────────┐ │
    ├─ -I──▼──┬─ipaddr───────────────┴─┤
    │         ├─ipaddr/prefixLen─────┤ │
    │         └─ipaddr/subnetmask────┘ │
    │         ┌───────────┐            │
    ├─ -L──▼────luname────┴────────────┤
    │         ┌─────────────┐          │
    ├─ -N──▼────applname────┴──────────┤
    │         ┌────────────┐           │
    └─ -P──▼────portnum────┴───────────┘
```

Filter description:

APPLname *applname*
Filter the output of the TELnet/**-t** report using the specified VTAM application name *applname*. You can enter up to six filter values and each specified value can be up to eight characters long.

CLIent/-E *clientname*
Filter the output of the TELnet/**-t** report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be up to eight characters long.

HOSTName/-H *hostname*

Filter the output of the TELnet/**-t** report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters long.

Result: At the end of the report, Netstat displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value, depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the TELnet/**-t** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength

Filter the output of the TELnet/**-t** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the TELnet/**-t** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* is the remote IP address.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as its IPv4 address.

Restrictions:

1. The filter value for an IPv6 address does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.

IPPort/-B *ipaddr+portnum*

Filter the report output of the TELnet/**-t** report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65 535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be either the local or remote IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.

LUName *luname*

Filter the output of the TELnet/**-t** report using the specified LU name *luname*. You can enter up to six filter values and each specified value can be up to eight characters long.

POrt/-P *portnum*

Filter the output of the TELnet/**-t** report using the specified port number *portnum*. You can enter up to six filter values.

Except for POrt/**-P**, and HOSTname/**-H**, the filter value can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a question mark (?), which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/**-I** filter, you must specify the value in the *ipaddr* format. The wildcard character is not accepted for the *ipaddr/prefixlen* or *ipaddr/subnetmask* format of IPAddr/**-I** values.

When you use z/OS UNIX **netstat**/**onetstat** command in a z/OS UNIX shell environment, take care if you use a z/OS UNIX MVS special character in a character string. It might cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, surround the character string with single quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the **-I** filter, issue the command as: **netstat -t -I '10.*.0.0'**.

Command syntax examples:

From TSO environment:

```
NETSTAT TELNET
   Display the status of the internal Telnet server connections in the default
   TCP/IP stack.
NETSTAT TELNET TCP TCPCS6
   Display the status of the internal Telnet server connections in TCPCS6 stack.
NETSTAT TELNET TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
   Display the status of the internal Telnet server connetctions in TCPCS8 stack
   whose foreign IP addresses match the specified filter IP address values.
NETSTAT TELNET (PORT 2222 6666 88
   Display the status of the internal Telnet server connections in the default
   TCP/IP stack whose foreign ports match the specified filter port numbers.
```

From UNIX shell environment:

```
   netstat -t
   netstat -t -p tcpcs6
   netstat -t -p tcpcs6 -I 9.43.1.1 9.43.2.2
   netstat -t -P 2222 6666 88
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT TELNET

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:41:00
Internal Telnet Server Status:
Conn     Foreign Socket         State    BytesIn  BytesOut ApplName LuName
----     --------------         -----    -------  -------- -------- ------
000000F6 201.2.10.11..1034      Establsh 00000715 00007648 TSO10002 TCPM1001
000000F9 201.2.10.12..1035      Establsh 00000222 00005930 TSO10004 TCPM1002
000000FE 9.27.11.182..4665      Establsh 00000091 00000623 TSO10003 TCPM1003
```

```
NETSTAT TELNET DETAIL

MVS TCP/IP NETSTAT CS V2R1       TCPIP NAME: TCPCS            17:41:00
Internal Telnet Server Status:
Conn     Foreign Socket         State    BytesIn  BytesOut ApplName LuName
----     --------------         -----    -------  -------- -------- ------
000000F6 201.2.10.11..1034      Establsh 00000715 00007648 TSO10002 TCPM1001
  ModeName: NSX32702  TnProto:  TN3270    TnUserId:
000000F9 201.2.10.12..1035      Establsh 00000222 00005930 TSO10004 TCPM1002
  ModeName: NSX32702  TnProto:  TN3270    TnUserId:
000000FE 9.27.11.182..4665      Establsh 00000091 00000623 TSO10003 TCPM1003
  ModeName: INTERACT  TnProto:  LINEMODE  TnUserId:
```

Note: For NETSTAT TELnet display, the BytesOut and BytesIn counts are in two forms:

nnnnnnnn
: Number range 0 – 99 999 999

nnnnnnnK
: Number range 100 000 000 – 4 294 967 294

  where *K* = *nnnnnnn* x 1000

IPv6 enabled or request for LONG format:

```
NETSTAT TELNET

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:11:25
Internal Telnet Server Status:
Conn     State    BytesIn    BytesOut   ApplName LuName
----     -----    -------    --------   -------- ------
000000F6 Establsh 0000000715 0000007648 TSO10002 TCPM1001
  Foreign socket: 201.2.10.11..1034
000000F9 Establsh 0000000222 0000005930 TSO10004 TCPM1002
  Foreign socket: 201.2.10.12..1035
000000FE Establsh 0000000091 0000000623 TSO10003 TCPM1003
  Foreign socket: 9.27.11.182..4665
```

```
NETSTAT TELNET DETAIL

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           11:11:25
Internal Telnet Server Status:
Conn     State    BytesIn    BytesOut   ApplName LuName
----     -----    -------    --------   -------- ------
000000F6 Establsh 0000000715 0000007648 TSO10002 TCPM1001
  Foreign socket: 201.2.10.11..1034
  ModeName: NSX32702  TnProto:  TN3270    TnUserId:
000000F9 Establsh 0000000222 0000005930 TSO10004 TCPM1002
  Foreign socket: 201.2.10.12..1035
  ModeName: NSX32702  TnProto:  TN3270    TnUserId:
000000FE Establsh 0000000091 0000000623 TSO10003 TCPM1003
  Foreign socket: 9.27.11.182..4665
  ModeName: INTERACT  TnProto:  LINEMODE  TnUserId:
```

For the NETSTAT TELnet display, the BytesOut and BytesIn counts are in one of the following five forms:

nnnnnnnnnn
: A number in the range 0 – 9 999 999 999

nnnnnnnnnK
: A number in the range 10 000 000 000 – 9 999 999 999 499 (*K* = *nnnnnnnnnn* x 1000)

nnnnnnnnnM
: A number in the range 9 999 999 999 500 – 9 999 999 999 499 999 (*M* = *nnnnnnnnnn* x 1 000 000)

nnnnnnnnnG
: A number in the range 9 999 999 999 500 000 – 9 999 999 999 499 999 999 (*G* = *nnnnnnnnnn* x 1 000 000 000)

nnnnnnnnnT
: A number in the range 9 999 999 999 500 000 000 – 9 999 999 999 499 999 999 999 (*T* = *nnnnnnnnnn* x 1 000 000 000 000)

Report field descriptions:

Conn The connection ID as it is known to TCP/IP. See Client ID or Connection Number information in "Netstat report general concept" on page 324 for a detailed description.

Foreign Socket
: See the Foreign Socket information in "Netstat report general concept" on page 324 for a detailed description.

State The connection state as it is known to TCP/IP. See the TCP connection status information in “Netstat report general concept” on page 324 for a detailed description.

BytesIn
Total bytes of data received from the client.

BytesOut
Total bytes of data sent to the client.

ApplName
The name of the application in session with the client.

LuName
The LU name selected by Telnet to represent the client.

ModeName
The SNA logmode used for this session.

InProto
The Telnet connection protocol used.

TN3270
The connection has negotiated to a TN3270 Telnet protocol.

TN3270E
The connection has negotiated to a TN3270E Telnet protocol.

TCLMODE
The connection has negotiated to a linemode Telnet protocol.

TnUserId
The user ID used specified on the solicitor panel in response to the Telnet request for user ID/password because of Restrictappl being coded. If an application user ID was entered on the solicitor panel, it is displayed in the TnUserId field. Otherwise, the TnUserId field is blank.

Netstat TTLS/-x report

Displays Application Transparent Transport Layer Security (AT-TLS) information. AT-TLS supports only TCP protocol connections.

TSO syntax:

```
►►──NETSTAT──TTLS──┤ Modifier ├──┤ Target ├──┤ Output ├──────────────────►◄
```

Modifier:

```
      ┌─GRoup──────────────────┐
►►────┼────────────────────────┼──────────────────────────────────────────►◄
      ├─COnn──connid─┬────────┬┤
      │              └─DETAIL─┘│
      └─GRoup─┬────────┬───────┘
              └─DETAIL─┘
```

COnn *connid*
Displays AT-TLS information for the specified connection. This information includes the name of the AT-TLS policy rule and the names of the associated actions. The specified *connid* value is a number assigned by the TCP/IP stack to uniquely identify a socket entity. You can determine the *connid* from the Conn column in the Netstat ALLCOnn/**-a** report.

DETAIL

Displays the AT-TLS policy rule and the associated action details for the specified connection.

GRoup

Displays summary information for AT-TLS groups. AT-TLS groups are defined using the policy statement TTLSGroupAction. The AT-TLS group remains active as long as the TTLSGroupAction is current or there are active connections using the group.

DETAIL

Displays detailed information for AT-TLS groups.

Target:

Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

z/OS UNIX syntax:

```
►►──netstat── -x──┤ Modifier ├──┤ Target ├──┤ Output ├──────────────────────►◄
```

Modifier:

```
      ┌─GRoup─────────────────────┐
►►────┼───────────────────────────┼──────────────────────────────────────────►◄
      ├─COnn──connid──┬────────┬──┤
      │               └─DETAIL─┘  │
      └─GRoup──┬────────┬─────────┘
               └─DETAIL─┘
```

COnn *connid*

Displays AT-TLS information for the specified connection. This information includes the name of the AT-TLS policy rule and the names of the associated actions. The specified *connid* is a number assigned by the TCP/IP stack to uniquely identify a socket entity. You can determine the *connid* from the Conn column in the Netstat ALLCOnn/**-a** report.

DETAIL

Displays the AT-TLS policy rule and the associated action details for the specified connection.

GRoup

Displays summary information for AT-TLS groups. AT-TLS groups are defined using the policy statement TTLSGroupAction. The AT-TLS group remains active as long as the TTLSGroupAction is current or there are active connections using the group.

DETAIL

Displays detailed information for AT-TLS groups.

Target:

Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT TTLS   (defaults to NETSTAT TTLS GROUP)
NETSTAT TTLS CONN 1B TCP TCPCS8
   Display summary AT-TLS information for the specified connection in the TCPCS8
   stack.
NETSTAT TTLS CONN 1B DETAIL TCP TCPCS8
   Display detailed AT-TLS information for the specified connection in the TCPCS8
   stack.
NETSTAT TTLS GROUP
   Display summary information for active AT-TLS groups.
NETSTAT TTLS GROUP DETAIL
   Display detailed information for active AT-TLS groups.
```

From UNIX shell environment:

```
   netstat -x        (defaults to -x GROUP)
   netstat -x CONN 1b -p tcpcs8
   netstat -x CONN 1b DETAIL -p tcpcs8
   netstat -x GROUP
   netstat -x GROUP DETAIL
```

COnn report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           19:51:22
 ConnID: 000000B8
   JobName:      FTPD1
   LocalSocket:  ::ffff:127.0.0.1..21
   RemoteSocket: ::ffff:127.0.0.1..1030
   SecLevel:     TLS Version 1.2
   Cipher:       C001 TLS_ECDH_ECDSA_WITH_NULL_SHA
   CertUserID:   N/A
   MapType:      Primary
   FIPS140:      Off
 TTLSRule: ftp_serv_21
   TTLSGrpAction:  grp_act1
   TTLSEnvAction:  env_act_serv

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           19:51:53
ConnID: 000000B8
  JobName:      FTPD1
  LocalSocket:  ::ffff:127.0.0.1..21
  RemoteSocket: ::ffff:127.0.0.1..1030
  SecLevel:     TLS Version 1.2
  Cipher:       C001 TLS_ECDH_ECDSA_WITH_NULL_SHA
  CertUserID:   N/A
  MapType:      Primary
  FIPS140:      Off
TTLSRule: ftp_serv_21
  Priority:       1
  LocalAddr:      All
  LocalPort:      21
  LocalPort:      2021
  LocalPortFrom:  620                LocalPortTo:  621
  RemoteAddr:     All
  RemotePort:     All
  Direction:      Inbound
  TTLSGrpAction:  grp_act1
    GroupID:                    00000006
    GroupUserInstance:          6
    TTLSEnabled:                On
    Envfile:                    /tmp/grp1.env
    CtraceClearText:            On
    Trace:                      255
    SyslogFacility:             Daemon
    SecondaryMap:               Off
    FIPS140:                    Off
  TTLSEnvAction:  env_act_serv
    EnvironmentUserInstance:    8
    HandshakeRole:              Server
    Keyring:                    /u/user3/testdb
    KeyringPW:                  Yes
    V3CipherSuites:             C001 TLS_ECDH_ECDSA_WITH_NULL_SHA
                                C002 TLS_ECDH_ECDSA_WITH_RC4_128_S
                                     HA
                                C003 TLS_ECDH_ECDSA_WITH_3DES_EDE_
                                     CBC_SHA
                                C004 TLS_ECDH_ECDSA_WITH_AES_128_C
                                     BC_SHA
    CtraceClearText:            On
    Trace:                      255
    SSLV2:                      Off
    SSLV3:                      On
    TLSV1:                      On
    TLSV1.1:                    On
    TLSV1.2:                    On
    ResetCipherTimer:           0
    ApplicationControlled:      On
    HandshakeTimeout:           10
    CertificateLabel:           ecdh_ecdsa_secp384r1
    SecondaryMap:               On
    TruncatedHMAC:              Off
    ClientMaxSSLFragment:       Off
    ServerMaxSSLFragment:       Off
```

```
ClientHandshakeSNI:         Off
    ServerHandshakeSNI:         Off
    ClientAuthType:             Required
    CertValidationMode:         Any
Renegotiation:              Default
    RenegotiationIndicator:     Optional
    RenegotiationCertCheck:     Off
    SuiteBProfile:              Off
    GSK_CRL_CACHE_TIMEOUT:      0
```

Report field descriptions:

Result: A field in a policy rule or policy action is displayed only when a value was configured for that attribute or when the attribute has a default value. Fields which were left undefined and have no default value are not displayed.

ApplicationControlled

Indicates whether the owning application can request AT-TLS security for the connection using the SIOCTTLSCTL IOCTL call.

Result: For a particular connection, the ApplicationControlled value on the TTLSConnectionAction, if specified, overrides the ApplicationControlled value on the TTLSEnvironmentAction.

CertificateLabel

The label of the authentication key used for the connection.

Result: For a particular connection, the CertificateLabel value on the TTLSConnectionAction statement, if specified, overrides the CertificateLabel value on the TTLSEnvironmentAction statement. If a CertificateLabel value is not specified on either the TTLSConnectionAction statement or the TTLSEnvironmentAction statement, the keyring default certificate is used.

CertUserID

The user ID, if any, that is associated with the partner's certificate. If no associated user ID is available, `N/A` is displayed.

CertValidationMode

The method of certificate validation that is in use for this connection. See the following possible values:

- The default value, `any`, means that any supported X.509 certificate validation method can be used.
- `RFC2459`, indicates that certificates are validated using the method described in RFC2459.
- `RFC3280`, indicates that certificates are validated using the method described in RFC3280.

Cipher

The cipher currently in use for encryption and decryption of data for the connection.

ClientAuthType

The level of Client Authentication used when the HandshakeRole is set to a value of ServerWithClientAuth. See the following possible values:

- The default value, *Required*, means that the client must present a certificate and that the certificate must pass verification.
- *PassThru* indicates that a certificate is not required and that no verification is attempted.

- *Full* indicates that the certificate is validated if the client presents one, but that the client is not required to present one.
- *SAFCheck* indicates that the client must present a certificate that must pass validation and be associated with a user ID in the security product.

ClientECurves

The list of elliptic curves that are supported by the client in the sequence of preference for use. The elliptic curve specifications are used by the client to guide the server as to which elliptical curves can be used when using cipher suites that use elliptical curve cryptography for the TLSv1.0 and higher protocols. Both the four-character value of the elliptic curve and the constant of the elliptic curve name are shown for each member of the list. The curve name `Any` indicates that both Brainpool standard curves and NIST standard curves can be used.

ClientMaxSSLFragment

For TLSv1.0 and higher protocols, this field specifies the client level of support for client-specified SSL fragment size on the connection. See the following possible values:

- `Required` indicates that maximum SSL fragment size support must be accepted by the server. Connections are closed if the server does not support the maximum SSL fragment size extension.
- `Optional` indicates that maximum SSL fragment size support is used if the server supports the function, but connections to servers that do not support this extension are accepted.
- `Off` indicates that maximum SSL fragment size support is not available; the TLS extension is not enabled. If the server requires SSL fragment size support, the client will be unable to connect. This is the default.

Result: For a particular connection, the ClientMaxSSLFragment value on the TTLSConnectionAction statement, if specified, overrides the ClientMaxSSLFragment value on the TTLSEnvironmentAction statement.

ClientMaxSSLFragmentLength

For SSL clients that use TLSv1.0 or higher protocols, specifies the maximum SSL fragment size to request on the connection, in bytes. See the following possible values:

- 512
- 1024
- 2048
- 4096

Result: For a particular connection, the ClientMaxSSLFragmentLength value on the TTLSConnectionAction statement, if specified, overrides the ClientMaxSSLFragmentLength value on the TTLSEnvironmentAction statement.

ClientHandshakeSNI

For TLSv1.0 and higher protocols, this field specifies the client level of support for client-specified server names on the connection handshake. See the following possible values:

- `Required` indicates that client-specified server name support must be accepted by the server. Connections are closed if the server does not support the client-specified server name extension.

- `Optional` indicates that client-specified server name support is used if the server supports the function, but connections to servers that do not support this extension are accepted.
- `Off` indicates that client-specified server name support is not available; the TLS extension is not enabled. If the server requires client-specified server name support, the client is unable to connect. This is the default value.

Result: For a particular connection, the ClientHandshakeSNI value on the TTLSConnectionAction statement, if specified, overrides the ClientHandshakeSNI value on the TTLSEnvironmentAction statement.

ClientHandshakeSNIMatch
For SSL clients using TLSv1.0 or higher protocols that might negotiate server name indication, this field specifies the level at which the client requires the client-specified server name to match a server name in the list of names that is maintained by the TLS server. See the following possible values:

- `Required` indicates that a server name in the list of server names provided by the TLS client must match a server name in the server name/certificate label list at the TLS server. The connection ends if no match can be found for the server name.
- `optional` indicates that connections are allowed to continue if no match is found for the server name. This is the default value.

Result: For a particular connection, the ClientHandshakeSNIMatch value on the TTLSConnectionAction statement, if specified, overrides the ClientHandshakeSNIMatch value on the TTLSEnvironmentAction statement.

ClientHandshakeSNIList
For SSL clients using TLSv1.0 or higher protocols that might negotiate server name indication, specifies a server name or names the client will pass to the server.

Result: For a particular connection, the ClientHandshakeSNIList value on the TTLSConnectionAction statement, if specified, overrides the ClientHandshakeSNIList value on the TTLSEnvironmentAction statement.

ConnID
The TCP/IP stack defined unique connection ID representing the connection.

ConnectionUserInstance
The instance identifier configured for the TTLSConnectionAction statement that is in use by the connection. The instance number can be used to signal a change without modifying other configuration statements. Valid values are in the range 0–65535.

CtraceClearText
Indicates whether application data traced for the connection, using Ctrace or datatrace, is shown as unencrypted data.

Result: For a particular connection, the CtraceClearText value on the TTLSConnectionAction statement, if specified, overrides the CtraceClearText value on the TTLSEnvironmentAction statement which, in turn, (if specified) overrides the CtraceClearText value on the TTLSGroupAction statement.

Direction

The connection direction condition specified in the policy rule that was mapped to the connection. See the following possible values:

- *Inbound* indicates that a connection request must arrive inbound to the local host to satisfy the rule.
- *Outbound* indicates that a connection request must be initiated by the local host to satisfy the rule.
- *Both* indicates that both Inbound and Outbound connection requests will match the rule.

The connection must match this condition.

Envfile

The name of the file that contains environment variables that are in use by the connection's language environment. The language environment was initialized with the CEE_ENVFILE environment variable set to this file. Environment variables such as CEE_RUNOPTS can be set in this file.

EnvironmentUserInstance

The instance identifier that is configured for the TTLSEnvironmentAction statement in use by the connection. The instance number can be used to signal a change without modifying other configuration statements. Valid values are in the range 0 – 65 535.

FIPS140

Indicates whether FIPS 140 support is enabled for the AT-TLS group to which the connection belongs.

GroupID

A value generated by AT-TLS that uniquely identifies the group of AT-TLS language environments (the AT-TLS group) to which the connection belongs.

GroupUserInstance

The instance identifier that is configured for the TTLSGroupAction statement in use by the connection. The instance number can be used to signal a change without modifying other configuration statements. Valid values are in the range 0 – 65 535.

GSK_CRL_CACHE_TIMEOUT

The certificate revocation list (CRL) cache timeout for the AT-TLS environment to which the connection belongs. This is the number of hours that a cached CRL remains valid. The value 0 indicates that CRL caching is disabled. See z/OS Cryptographic Services System SSL Programming for details.

GSK_CRL_SECURITY_LEVEL

The certificate revocation list (CRL) security level for the AT-TLS environment to which the connection belongs. See the following possible values:

- `Low` indicates that certificate validation does not fail if the LDAP server cannot be contacted.
- `Medium` indicates that certificate validation requires the LDAP server to be contactable, but does not require a CRL to be defined. This is the default
- `High` indicates that certificate validation requires the LDAP server to be contactable, and a CRL to be defined.

GSK_LDAP_SERVER
The LDAP server host names for the AT-TLS environment to which the connection belongs. Each name can contain an optional port number separated from the name by a colon. See z/OS Cryptographic Services System SSL Programming for details.

GSK_LDAP_SERVER_PORT
The LDAP server port for the AT-TLS environment to which the connection belongs. See z/OS Cryptographic Services System SSL Programming for details.

GSK_LDAP_USER
The distinguished name used when connecting to the LDAP server for the AT-TLS environment to which the connection belongs. See z/OS Cryptographic Services System SSL Programming for details.

GSK_LDAP_USER_PW
Indicates whether the AT-TLS environment to which the connection belongs uses a password when connecting to the LDAP server. See z/OS Cryptographic Services System SSL Programming for details.

GSK_SYSPLEX_SIDCACHE
Indicates whether sysplex session caching is enabled for the AT-TLS environment to which the connection belongs. See z/OS Cryptographic Services System SSL Programming for details.

GSK_V2_SESSION_TIMEOUT
The SSL version 2 session timeout for the AT-TLS environment to which the connection belongs. This is the number of seconds until a session identifier expires. See z/OS Cryptographic Services System SSL Programming for details.

GSK_V2_SIDCACHE_SIZE
The size of the SSL version 2 session identifier cache for an AT-TLS environment. See z/OS Cryptographic Services System SSL Programming for details.

GSK_V3_SESSION_TIMEOUT
The SSL version 3 or TLS version 1 session timeout for an AT-TLS environment. This is the number of seconds until a session identifier expires. See z/OS Cryptographic Services System SSL Programming for details.

GSK_V3_SIDCACHE_SIZE
The size of the SSL version 3 or TLS version 1 session identifier cache for an AT-TLS environment. See z/OS Cryptographic Services System SSL Programming for details.

HandshakeTimeout
The number of seconds that the connection waits for the initial handshake to complete. Valid values are in the range 0 – 600.

For connections with HandshakeRole set to Client, the timer is initially set to 5 times this value, allowing for network delay and any delay on the server in processing the connection. When the initial response is received from the server, the timer is reset to this value so that the initial handshake can complete.

For connections with HandshakeRole set to Server or ServerWithClientAuth, when the server starts to process the new connection, the timer is set to this value and the server then waits for the initial request from the client. When the server sends the initial response, the timer is reset to this value so that the initial handshake can complete.

If the timer expires, the TCP connection is reset. A value of 0 indicates that the connection does not time out waiting for the initial handshake to complete.

Result: For a particular connection the HandshakeTimeout value on the TTLSConnectionAction, if specified, overrides the HandshakeTimeout value on the TTLSEnvironmentAction.

HandshakeRole
The SSL handshake role for the connection. See the following possible values:

- *Client* indicates that the handshake is to be performed as a client.
- *Server* indicates that the handshake is to be performed as a server.
- *ServerWithClientAuth* indicates that the handshake is to be performed as a server requiring client authentication.

Result: For a particular connection, the HandshakeRole value on the TTLSConnectionAction, if specified, overrides the HandshakeRole value on the TTLSEnvironmentAction statement.

JobName
When part of the ConnID section, the JobName value is the procedure name of the local application.

When part of the TTLSRule section, the JobName value is the job name condition that was specified in the policy rule that was mapped to the connection. If no JobName value is specified for a policy rule, all job names is the default. If specified, the connection must match this condition. A trailing asterisk indicates a wildcard specification.

Keyring
The path and file name of the key database z/OS UNIX file, z/OS PKCS #11 token name, or the RACF ring name for the AT-TLS environment to which the connection belongs.

KeyringPw
Indicates whether a z/OS UNIX file system key database password was configured for the AT-TLS environment to which the connection belongs.

KeyringStashFile
The path and file name of the z/OS UNIX file system key database password stash file for the AT-TLS environment to which the connection belongs.

LocalAddr
A single local IP address (or a range of local IP addresses when the range was configured using the format ipv4_addr/num_mask_bits or the format ipv6_addr/prefixLength) that is a condition specified in the policy rule that was mapped to the connection. If specified, the connection must match this condition.

- If `0.0.0.0/0` is specified, this rule applies to all IPv4 addresses.
- If `::/0` is specified, the rule applies to all IPv6 addresses.
- If `All` is displayed, any address matches this condition.

LocalAddrFrom/LocalAddrTo
A range of local IP addresses, when the range was configured using a start and end address pair, that is a condition specified in the policy rule that

was mapped to the connection. If neither LocalAddr nor LocalAddrFrom/LocalAddrTo is specified, all addresses is the default. If specified, the connection must match this condition.

LocalPort
A single local port that is a condition specified in the policy rule that was mapped to the connection. If specified, the connection must match this condition. If All is displayed, any port matches this condition.

LocalPortFrom/LocalPortTo
A range of local ports, configured using a start and end pair, that is a condition specified in the policy rule that was mapped to the connection. If neither LocalPort nor LocalPortFrom/LocalPortTo is specified, all ports is the default. If specified, the connection must match this condition.

LocalSocket
The local socket of the connection. See the Local Socket information "Netstat report general concept" on page 324 for a detailed description.

MapType
The mapping method used to locate this policy. See the following possible values:

- `Primary` indicates that this connection matched the rule conditions of the indicated policy rule.
- `Secondary` indicates that this connection was established between the same two IP addresses by the same process that has a connection that used the primary mapping method to locate this policy that has SecondaryMap set **On**.

Priority
The priority associated with the policy rule that was mapped to the connection. A higher priority value indicates a higher priority rule. Priority can be used to differentiate between rules when a connection could match more than one of the configured rules. Valid values are in the range 1–255. The default value is 0.

RemoteAddr
A single remote IP address (or a range of remote IP addresses when the range was configured using the format ipv4_addr/num_mask_bits or the format ipv6_addr/prefixLength) that is a condition specified in the policy rule that was mapped to the connection. If specified, the connection must match this condition.

- If `0.0.0.0/0` is specified, this rule applies to all IPv4 addresses.
- If `::/0` is specified, the rule applies to all IPv6 addresses.
- If `All` is displayed, any address matches this condition.

RemoteAddrFrom/RemoteAddrTo
A range of remote IP addresses, configured using a start and end address pair, that is a condition specified in the policy rule that was mapped to the connection. If neither RemoteAddr nor RemoteAddrFrom/RemoteAddrTo is specified, all addresses is the default. If specified, the connection must match this condition.

RemotePort
A single remote port that is a condition specified in the policy rule that was mapped to the connection. If specified, the connection must match this condition. If `All` is displayed, any port matches this condition.

RemotePortFrom/RemotePortTo
A range of remote ports, configured using a start and end pair, that is a condition specified in the policy rule that was mapped to the connection. If neither RemotePort nor RemotePortFrom/RemotePortTo is specified, all ports is the default. If specified, the connection must match this condition.

RemoteSocket
The remote socket of the connection. See the Foreign Socket information in "Netstat report general concept" on page 324 for a detailed description.

Renegotiation
The type of the session key renegotiation that is allowed by server.

Default
Indicates that SSLv3 and TLS handshake renegotiation is disabled and RFC 5746 renegotiation is enabled.

Disable
Indicates that all renegotiation is disabled.

Abbreviated
Indicates that SSLv3 and TLS abbreviated handshake renegotiation only for the current session is allowed; SSLv3 and TLS full handshake renegotiation is disabled; and RFC 5746 renegotiation is allowed.

All Indicates that all forms of renegotiation are allowed.

RenegotiationCertCheck
Indicates whether the peer's certificate is checked during renegotiation to prevent change to a different certificate.

RenegotiationIndicator
Indicates whether the partner must indicate the support for RFC 5746 renegotiation for initial handshake to proceed.

ResetCipherTimer
The number of minutes a secure connection can be active before a rehandshake is initiated by AT-TLS to establish a new session key for the connection. If not specified or specified as `0`, cipher reset is not initiated by AT-TLS. Valid values are in the range 0 – 1440.

Result: For a particular connection the ResetCipherTimer value on the TTLSConnectionAction statement, if specified, overrides the ResetCipherTimer value on the TTLSEnvironmentAction statement.

SecLevel
The security level being used by the connection: SSL Version 2, SSL Version 3, or TLS Version 1.

SecondaryMap
Indicates whether the application establishes secondary connections using dynamic port numbers. If so, the primary connection maps to this policy using rule conditions. Subsequent connections established by the same process between the same two IP addresses that do not map to their own policy or map to a policy with a lower priority than the primary connection are considered secondary connections. Secondary connections use the same policy as the associated primary connection.

ServerMaxSSLFragment

For TLSv1.0 and higher protocols, this field specifies the server level of support for client-specified SSL fragment size on the connection. See the following possible values:

- `Required` indicates that maximum SSL fragment size support must be accepted by the client. Connections are closed if the client does not support the maximum SSL fragment size extension.
- `Optional` indicates that maximum SSL fragment size support is used if the client supports the function, but connections to clients that do not support this extension are accepted.
- `Off` indicates that maximum SSL fragment size support is not available; the TLS extension is not enabled. If the client requires SSL fragment size support, the client is unable to connect. This is the default value.

  Result: For a particular connection, the ServerMaxSSLFragment value on the TTLSConnectionAction statement, if specified, overrides the MaximumSSLFragment value on the TTLSEnvironmentAction statement.

ServerHandshakeSNI

For TLSv1.0 and higher protocols, this field specifies the server level of support for client-specified server names on the connection handshake. See the following possible values:

- `Required` indicates that client-specified server name support must be accepted by the client. Connections are closed if the client does not support the client-specified server name extension.
- `Optional` indicates that client-specified server name support is used if the client supports the function, but connections to clients that do not support this extension are accepted.
- `Off` indicates that client-specified server name support is not available; the TLS extension is not enabled. If the client requires client-specified server name support, the client is unable to connect. This is the default value.

Result: For a particular connection, the ServerHandshakeSNI value on the TTLSConnectionAction statement, if specified, overrides the ServerHandshakeSNI value on the TTLSEnvironmentAction statement.

ServerHandshakeSNIMatch

For SSL servers that are using TLSv1.0 or higher protocols that might negotiate server name indication, this field specifies the level at which the server requires the client-specified server name to match a server name in the list of names that is maintained by the TLS server. See the following possible values:

- `Required` indicates that a server name in the list of server names that is provided by the TLS client must match a server name in the server name and certificate label list at the TLS server. The connection ends if no match can be found for the server name.
- `optional` indicates that connections are allowed to continue if no match is found for the server name. This is the default.

Result: For a particular connection, the ServerHandshakeSNIMatch value on the TTLSConnectionAction statement, if specified, overrides the ServerHandshakeSNIMatch value on the TTLSEnvironmentAction statement.

ServerHandshakeSNIList

For SSL servers that use TLSv1.0 or higher protocols that might negotiate server name indication, specifies server name and certificate label pairs to be used by the server when matching a name from the client.

Result: For a particular connection, the ServerHandshakeSNIList value on the TTLSConnectionAction statement, if specified, overrides the ServerHandshakeSNIList value on the TTLSEnvironmentAction statement.

SignaturePairs
The list of the pairs of TLSv1.2 signature and hash algorithm that are sent from the client to the server to indicate which pairs can be used in digital signatures of the server certificate. This field is ignored by servers that do not support TLSv1.2. Both the four-character value of signature/hash algorithm and the constant of the signature/hash algorithm name are shown for each member of the list.

SSLV2 Indicates whether SSL version 2 protocol is acceptable for the connection.

Result: For a particular connection the SSLV2 value on the TTLSConnectionAction statement, if specified, overrides the SSLV2 value on the TTLSEnvironmentAction statement.

SSLV3 Indicates whether SSL version 3 protocol is acceptable for the connection.

Result: For a particular connection the SSLV3 value on the TTLSConnectionAction statement, if specified, overrides the SSLV3 value on the TTLSEnvironmentAction statement.

SuiteBProfile
Indicates whether a Suite B profile of cipher suites should be used.

SyslogFacility
The syslog facility name this group uses when writing records to syslogd.

TLSV1
Indicates whether TLS version 1.0 protocol is acceptable for the connection.

Result: For a particular connection, the TLSV1 value on the TTLSConnectionAction statement, if specified, overrides the TLSV1 value on the TTLSEnvironmentAction statement.

TLSV1.1
Indicates whether TLS version 1.1 protocol is acceptable for the connection.

Result: For a particular connection the TLSV1.1 value on the TTLSConnectionAction statement, if specified, overrides the TLSV1.1 value on the TTLSEnvironmentAction statement.

TLSV1.2
Indicates whether TLS version 1.2 protocol is acceptable for the connection.

Result: For a particular connection, the TLSV1.2 value on the TTLSConnectionAction statement, if specified, overrides the TLSV1.2 value on the TTLSEnvironmentAction statement.

TruncatedHMAC
Indicates whether clients and servers can negotiate the use of 80-bit truncated MAC addresses. See the following possible values:

- `Required` indicates that 80-bit truncated MAC addresses must be accepted by both endpoints.
- `Optional` indicates that the use of 80-bit truncated MAC addresses is negotiated.

- `Off` indicates that 80-bit truncated MAC addresses are not supported. This is the default.

Result: For a particular connection, the TruncatedHMAC value on the TTLSConnectionAction statement, if specified, overrides the TruncatedHMAC value on the TTLSEnvironmentAction statement.

TTLSConnAction
The name of the policy action used to specify attribute differences between what is required for the connection and what is specified for the AT-TLS environment to which the connection belongs. This name was configured to Policy Agent using the TTLSConnectionAction statement. The name is followed by `(Stale)` when the action is no longer available for use by new connections.

TTLSEnabled
Indicates whether AT-TLS services are used by the connection.

TTLSEnvAction
The name of the policy action used to specify attributes for the AT-TLS environment to which the connection belongs. This name was configured to Policy Agent using the TTLSEnvironmentAction statement. The name is followed by `(Stale)` when the action is no longer available for use by new connections.

TTLSGrpAction
The name of the policy action used to specify attributes for the AT-TLS group to which the connection belongs. This name was configured to Policy Agent using the TTLSGroupAction statement.

- The name is followed by `(Stale)` when the action is no longer available for use by new connections.
- The name is followed by `(Failed)` if the group failed to initialize properly or experienced an unrecoverable abend.

TTLSRule
The name of the policy rule, configured to Policy Agent using the TTLSRule statement, that was mapped to the connection. For connections that match a rule, the determination of whether to use AT-TLS for the connection and how AT-TLS attributes are set when AT-TLS is used are determined by the policy actions associated with the policy rule. The name is followed by `(Stale)` when the rule is no longer available for use by new connections.

Trace The level of AT-TLS tracing for the connection.

Result: For a particular connection the Trace value on the TTLSConnectionAction, if specified, overrides the Trace value on the TTLSEnvironmentAction statement which in turn, if specified, overrides the Trace value on the TTLSGroupAction statement.

The level of tracing is a sum of the following numbers:

0 No tracing is enabled.

1 Error - Errors are traced to the TCP/IP job log.

2 Error - Errors are traced to syslogd. This is the default.

4 Info - Tracing of when a connection is mapped to an AT-TLS rule (and when a secure connection is successfully initiated) is enabled.

8 Event - Tracing of major events is enabled.

16 Flow - Tracing of system SSL calls is enabled.

32 Data - Tracing of encrypted negotiation is enabled. This traces the negotiation of secure sessions.

255 All tracing is enabled.

UserID
The application user ID condition specified in the policy rule that was mapped to the connection. A trailing asterisk indicates a wildcard specification. If not specified, all user IDs is the default. If specified, the connection must match this condition.

V2CipherSuites
The SSL version 2 cipher suite list (also known as cipher specifications), in order of preference, to be used for the connection. See gsk_environment_open() in z/OS Cryptographic Services System SSL Programming for a list of valid cipher specifications.

Result: For a particular connection the V2CipherSuites value on the TTLSConnectionAction statement, if specified, overrides the V2CipherSuites value on the TTLSEnvironmentAction statement.

V3CipherSuites
The SSL version 3 or TLS version 1 cipher suite list (also known as cipher specifications), in order of preference, to be used for the connection. Both the four-character value of the cipher and the constant of the cipher name are shown for each member of the list. See gsk_environment_open() in z/OS Cryptographic Services System SSL Programming for a list of valid cipher specifications.

Result: For a particular connection, the V3CipherSuites value on the TTLSConnectionAction statement, if specified, overrides the V3CipherSuites value on the TTLSEnvironmentAction statement.

Result: A field in a policy rule or policy action is displayed only when a value was configured for that attribute or when the attribute has a default value. Fields that were left undefined and have no default value are not displayed.

Group report examples:

NETSTAT TTLS GROUP

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           12:55:20
TTLSGrpAction                             Group ID           Conns
----------------------------------------  -----------------  -----
TTLSGrpAction15 (Stale)                   00000004              25
TTLSGrpAction5                            00000007 (Failed)      0
```

NETSTAT TTLS GROUP DETAIL

```
MVS TCP/IP NETSTAT CS V2R1      TCPIP Name: TCPCS           12:55:20
TTLSGrpAction:   TTLSGrpAction15 (Stale)
  GroupID:         00000004
  Tasks:           10                   GroupConns:      25
  WorkQElements:   7                    SyslogQElements: 1
    Env: TTLSEnvAction9                            EnvConns: 25
TTLSGrpAction:   TTLSGrpAction5
  GroupID:         00000007 (Failed)
  Tasks:           0                    GroupConns:      0
  WorkQElements:   0                    SyslogQElements: 0
```

Report field descriptions:

EnvConns
The number of connections currently created within the AT-TLS environment.

GroupConns
The number of connections currently created within the AT-TLS group.

GroupID
A value generated by AT-TLS that uniquely identifies a group of AT-TLS language environments (an AT-TLS group) in a TCP/IP stack.

SyslogQElements
The number of AT-TLS tracing work elements waiting to be processed in the group.

Tasks The number of MVS tasks currently allocated to support the AT-TLS work in the group.

Env The name of a policy action used to specify attributes for an AT-TLS environment. This name was configured to Policy Agent using the TTLSEnvironmentAction statement. The name is followed by `(Stale)` when the action is no longer available for use by new connections.

TTLSGrpAction
The name of a policy action used to specify attributes for a group of AT-TLS environments. This name was configured to Policy Agent using the TTLSGroupAction statement. The name is followed by `(Stale)` when the action is no longer available for use by new connections.

WorkQElements
The number of work elements waiting to be processed in the group.

Netstat Up/-u report

Displays the date and time that TCP/IP was started and specifies whether it is IPv6 enabled or disabled.

TSO syntax:

```
►►──NETSTAT Up──┤ Target ├──┤ Output ├──────────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

z/OS UNIX syntax:

```
►►──netstat -u──┤ Target ├──┤ Output ├──────────────────────────────────►◄
```

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT UP
   Display the date and time that TCP/IP was started and specifies whether it is
   IPv6 enabled or disabled for the default TCP/IP stack.
NETSTAT UP TCP TCPCS6
   Display the date and time that TCP/IP was started and specifies whether it is
   IPv6 enabled or disabled for the TCPCS6 stack.
```

From UNIX shell environment:

```
   netstat -u
   netstat -u -p tcpcs6
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT UP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:34:37
Tcpip started at 14:27:29 on 01/31/2002 with IPv6 disabled
```

IPv6 enabled or request for LONG format:

```
NETSTAT UP
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:34:37
Tcpip started at 14:27:29 on 01/31/2002 with IPv6 enabled
```

Netstat VCRT/-V report

Displays the dynamic VIPA Connection Routing Table used for sysplex distributor and moveable dynamic VIPA support. On a sysplex distributor routing stack, it displays all connections being routed through the distributor. On a stack taking over a dynamic VIPA, it displays every connection to the dynamic VIPA. On a sysplex distributor target stack or a stack that is in the process of giving up a dynamic VIPA, the report displays every connection for which the stack is an endpoint.

TSO syntax:

```
►►──NETSTAT VCRT──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────►◄
```

Modifier:

```
►►──DETAIL─────────────────────────────────────────────────────────────────────►◄
```

DETAIL

Displays the general dynamic VIPA Connection Routing Table information plus the following additional information for each connection:

- Policy rule and policy action.
- Timed affinity-related information.
- Indicates whether the connection is eligible for acceleration. This information is not displayed if the stack is not enabled for acceleration. For details about how to enable a stack for acceleration, see the IPCONFIG profile statement information in the z/OS Communications Server: IP Configuration Reference.
- Information about the route used by the stack which owns a dynamic VIPA to send packets to the target stack. This information is not displayed on a target stack or if no VIPAROUTE profile statements have been configured to the stack. The routing information provided describes the route used in forwarding the last packet received for this connection to the target stack.

  The routing information might describe the best available route to reach the IP address, which was defined in the VIPAROUTE statement for that target stack, or it might describe the dynamic XCF route for that target stack. See VIPADYNAMIC information in the z/OS Communications Server: IP Configuration Reference for details about the VIPAROUTE statement. For more information about the use of the routing information, see route selection for distributing packets details in the z/OS Communications Server: IP Configuration Guide.

Target:

Provides the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:

```
►►──┬─HOSTName──hostname───────────────────┬──────────────────────────►◄
    │           ┌─────────────────────┐    │
    │           ▼                     │    │
    ├─IPAddr──────┬─ipaddr────────────┬─┴────┤
    │             ├─ipaddr/prefixLen──┤      │
    │             └─ipaddr/subnetmask─┘      │
    │           ┌────────────────┐           │
    │           ▼                │           │
    ├─IPPort──────ipaddr+portnum─┴───────────┤
    │        ┌──────────────┐                │
    │        ▼              │                │
    └─POrt────────portnum───┴────────────────┘
```

z/OS UNIX syntax:

```
►►──netstat -V──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────────►◄
```

Modifier:

```
▶▶──DETAIL──────────────────────────────────────────────────────────▶◀
```

DETAIL

Displays the general dynamic VIPA Connection Routing Table information plus the following additional information for each connection:

- Policy rule and policy action.
- Timed affinity-related information.
- Indicates whether the connection is eligible for acceleration. This information is not displayed if the stack is not enabled for acceleration. For details about how to enable a stack for acceleration, see the IPCONFIG profile statement information in the z/OS Communications Server: IP Configuration Reference.
- Information about the route used by the stack which owns a dynamic VIPA to send packets to the target stack. This information is not displayed on a target stack or if no VIPAROUTE profile statements have been configured to the stack. The routing information provided describes the route used in forwarding the last packet received for this connection to the target stack.

  The routing information might describe the best available route to reach the IP address, which was defined in the VIPAROUTE statement for that target stack, or it might describe the dynamic XCF route for that target stack. See VIPADYNAMIC details in the z/OS Communications Server: IP Configuration Reference for information about the VIPAROUTE statement. For more details about the use of the routing information, see information about route selection for distributing packets in the z/OS Communications Server: IP Configuration Guide.

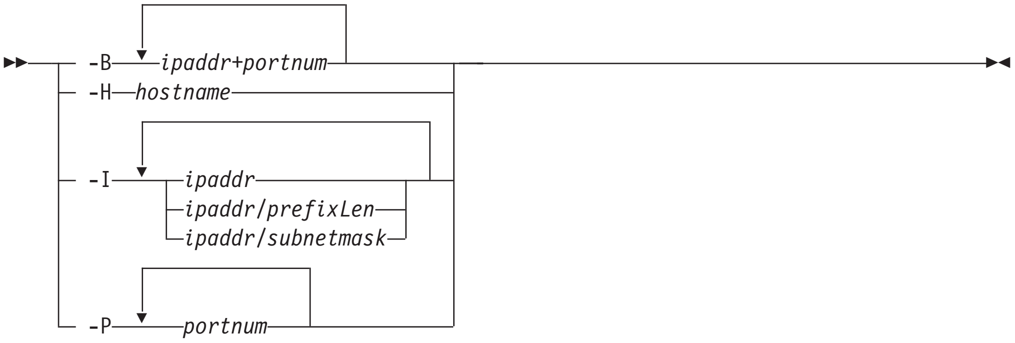
Target:

Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:

The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:



Filter description:

HOSTName/-H *hostname*

Filter the output of the VCRT/**-V** report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 255 characters long.

Result: At the end of the report, Netstat displays the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver that it used as filters.

Restrictions:

1. The HOSTName/**-H** filter does not support wildcard characters.
2. Using the HOSTName/**-H** filter might cause delays in the output due to resolution of the *hostname* value, depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the VCRT/**-V** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength

Filter the output of the VCRT/**-V** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the VCRT/**-V** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be a source IP address, a destination IP address, or a destination XCF IP address.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and will usually provide the same result as its IPv4 address.

Restrictions:

1. The IPAddr/**-I** option for VCRT/**-V** report does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

IPPort/-B *ipaddr+portnum*

Filter the report output of the VCRT/**-V** report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0–65535. The filter values *ipaddr* and *portnum* matches any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be a source IP address, a destination IP address, or a destination XCF IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.
- For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.

POrt/-P *portnum*

Filter the output of the VCRT/**-V** report using the specified port number *portnum*. You can enter up to six filter values.

Guideline: The port number can be either a local or remote port.

Restriction: For a UDP endpoint socket, the filter value applies only to the local or source port.

Command syntax examples:

From TSO environment:

```
NETSTAT VCRT
   Displays the dynamic VIPA Connection Routing Table information in the default
   TCP/IP stack.
NETSTAT VCRT TCP TCPCS6
   Displays the dynamic VIPA Connection Routing Table information in the TCPCS6
   stack.
```

From UNIX shell environment:

```
   netstat -V
   netstat -V -p tcpcs6
```

Report examples:

The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT VCRT

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           18:17:26
Dynamic VIPA Connection Routing Table:
Dest IPaddr      DPort  Src IPaddr       SPort  DestXCF Addr
-----------      -----  ----------       -----  ------------
201.2.10.11      00021  193.9.200.1      00000  193.1.1.18
201.2.10.11      00021  193.9.200.1      01025  193.1.1.18
201.2.10.11      00021  201.1.10.85      01026  201.1.10.10
203.1.10.18      08000  193.10.1.1.118   01080  193.1.1.108
```

```
NETSTAT VCRT DETAIL

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           14:16:16
Dynamic VIPA Connection Routing Table:
Dest IPaddr      DPort  Src IPaddr       SPort  DestXCF Addr
-----------      -----  ----------       -----  ------------
201.2.10.11      00021  201.1.10.85      00000  201.1.10.10
  CfgTimAff: 0200  TimAffCnt: 0000000002  TimAffLft: 0000
201.2.10.11      00021  201.1.10.85      01026  201.1.10.10
  PolicyRule:    *NONE*
  PolicyAction:  *NONE*
  Intf:  CTC1
    VipaRoute: Yes      Gw: 0.0.0.0
  Accelerator: No
201.2.10.11      00021  201.1.10.85      01027  201.1.10.10
  PolicyRule:    *NONE*
  PolicyAction:  *NONE*
  Intf:   OSAQDIOLINK
    VipaRoute: Yes      Gw: 199.100.1.1
  Accelerator: yes
203.1.10.18      08000  193.10.1.118     01080  193.1.1.108
  PolicyRule:    PRule-TCP-High
  PolicyAction:  PAction-TCP-High
  Intf:   EZAXCFC7
    VipaRoute: No       Gw: 0.0.0.0
  Accelerator: No
203.1.10.19      09000  193.10.1.119     01081  193.1.1.109
  PolicyRule:    PRule-TCP-High
  PolicyAction:  PAction-TCP-High
  Intf:   EZAXCFC6
    VipaRoute: Unavail  Gw: 0.0.0.0
  Accelerator: No
```

IPv6 enabled or request for LONG format:

NETSTAT VCRT

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:04:04
Dynamic VIPA Connection Routing Table:
Dest:      201.2.10.11..21
  Source:  193.9.200.1..0
  DestXCF: 193.1.1.18
Dest:      201.2.10.11..21
  Source:  193.9.200.1..1025
  DestXCF: 193.1.1.18
Dest:      201.2.10.11..21
  Source:  201.1.10.85..1026
  DestXCF: 201.1.10.10
Dest:      203.1.10.18..8000
  Source:  193.9.200.1..1080
  DestXCF: 193.1.1.108

Dest:      2001:0db8::0522:f103..21
  Source:  2001:0db8::0524:f104..1026
  DestXCF: 2001:0db8::0943:f003
```

NETSTAT VCRT DETAIL

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           20:04:04
Dynamic VIPA Connection Routing Table:
Dest:      201.2.10.11..21
  Source:  201.1.10.85..0
  DestXCF: 201.1.10.10
    CfgTimAff: 0200  TimAffCnt: 0000000002  TimAffLft: 0000
Dest:      201.2.10.11..21
  Source:  201.1.10.85..1026
  DestXCF: 201.1.10.10
    PolicyRule:    *NONE*
    PolicyAction:  *NONE*
    Intf:   CTC1
      VipaRoute: Yes      Gw: 0.0.0.0
  Accelerator: No
Dest:      201.2.10.11..21
  Source:  201.1.10.85..1027
  DestXCF: 201.1.10.10
    PolicyRule:    *NONE*
    PolicyAction:  *NONE*
    Intf:   OSAQDIOLINK
      VipaRoute: Yes      Gw: 199.100.1.1
  Accelerator: No
Dest:      203.1.10.18..8000
  Source:  193.9.200.1..1080
  DestXCF: 193.1.1.108
    PolicyRule:    PRule-TCP-High
    PolicyAction:  PAction-TCP-High
    Intf:   EZAXCFC7
      VipaRoute: No       Gw: 0.0.0.0
  Accelerator: No
Dest:      203.1.10.19..9000
  Source:  193.9.10.119..1081
  DestXCF: 193.1.1.109
    PolicyRule:    PRule-TCP-High
    PolicyAction:  PAction-TCP-High
    Intf:   EZAXCFC6
      VipaRoute: Unavail  Gw: 0.0.0.0
  Accelerator: No
 Dest:     2ec0::0522:f103..21
  Source:  2ec0::0524:f104..1026
  DestXCF: 2ec0::0943:f003
    PolicyRule:    PRule-TCP-High
    PolicyAction:  PAction-TCP-High
    Intf:   OSAQDIO46
      VipaRoute: Yes      Gw:  2ec0::206:2aff:fe71:4400
```

Report field descriptions:

For a SHORT format report:

Dest IPaddr
The destination IP address for this connection.

DPort The destination port for this connection.

Src IPaddr
The source IP address for this connection. If the source IP address value is 0 for an entry, then the entry does not represent an established connection. Entries with a source IP address value 0 represent an affinity between a client IP address and a dynamic VIPA destination IP address and port. Such an affinity arises from passive-mode FTP. Each affinity entry is immediately followed by all the established connection entries that are associated with the affinity.

SPort The source port for this connection. If the source port value is 0 for an entry, then the entry does not represent an established connection. Entries with a source port value 0 represent an affinity between a client IP address and a dynamic VIPA destination IP address and port. Such an affinity might arise from passive-mode FTP or from a distributed DVIPA with a nonzero value for the TIMEDAFFINITY parameter on the VIPADISTRIBUTE profile statement. Each affinity entry is immediately followed by all the established connection entries that are associated with the affinity.

DestXCF Addr
The dynamic XCF address of the stack that is processing this connection. For connections to and from non-z/OS tier 1targets, this value is the IP address of the tier 1 target.

For a LONG format report:

Dest The destination IP address and port for this connection.

Source
The source IP address and port for this connection. If the source IP address value is 0 for an entry, then the entry does not represent an established connection. Entries with a source IP address value of zero represent an affinity between a client IP address and a dynamic VIPA destination IP address and port. Such an affinity arises from passive-mode FTP. Each affinity entry is immediately followed by all the established connection entries that are associated with the affinity.

If the source port value is zero for an entry, then the entry does not represent an established connection. Entries with a source port value of zero represent an affinity between a client IP address and a dynamic VIPA destination IP address and port. Such an affinity might arise from passive-mode FTP or from a distributed DVIPA with a nonzero value for the TIMEDAFFINITY parameter on the VIPADISTRIBUTE profile statement. Each affinity entry is immediately followed by all the established connection entries that are associated with the affinity.

DestXCF
The dynamic XCF address of the stack that is processing this connection. For connections to and from non-z/OS tier 1targets, this value is the IP address of the tier 1 target.

For a SHORT or LONG format report:

DETAIL

For each entry that represents an established dynamic VIPA connection or an affinity created by the passive-mode FTP, displays the preceding information plus the following policy rule and action.

PolicyRule

The policy rule name configured to the Policy Agent. The PolicyRule value `*NONE*` indicates that the connection was not mapped to a policy rule.

PolicyAction

The policy action name configured to the Policy Agent. A PolicyAction value `*NONE*` indicates that the connection was not mapped to a policy action.

For each entry that represents an established dynamic VIPA connection on the stack which owns the dynamic VIPA (when VIPAROUTE profile statements have been configured to the stack), displays the preceding information plus the following additional routing information.

Intf The name of the interface for the route being used to distribute packets to the target stack. The value `*NONE*` indicates that there is no route associated with this connection.

VipaRoute

Indicates whether the VIPAROUTE parameter is being used to route packets to the target stack for this connection:

No Indicates that the dynamic XCF interface is being used to distribute packets to the target stack.

Yes Indicates that the best available route, based on the VIPAROUTE parameters, is being used to distribute packets to the target stack.

Unavail

Indicates that the TCP/IP stack attempted to use the route based on the VIPAROUTE parameters, but an error was detected during the verification of the VIPAROUTE statement. Because of this, the dynamic XCF interface is being used to distribute packets to that target stack. See VIPADYNAMIC statement information in the z/OS Communications Server: IP Configuration Reference for details about the VIPAROUTE statement.

Gw The gateway used to send packets to the target stack. If the value is equal to 0.0.0.0 for an IPv4 entry or :: for an IPv6 entry, then the destination is directly reachable without needing to go through a gateway.

Accelerator

Indicates whether this connection is eligible for the QDIO Accelerator function. To be eligible, the QDIO accelerator function must be enabled by specifying the QDIOACCELERATOR parameter on the IPCONFIG statement. The accelerator field is displayed only if QDIOACCELERATOR is specified on the IPCONFIG statement. For more information about the IPCONFIG statement, see the IPCONFIG profile statement information in z/OS Communications Server: IP Configuration Reference

The packets that are eligible for acceleration are those that are received by the Sysplex Distributor and that are forwarded to a target stack in any of the following inbound and outbound DLC combinations:

- Inbound HiperSockets, forwarded outbound over OSA-Express QDIO connections
- Inbound OSA-Express QDIO, forwarded outbound over the dynamic XCF HiperSockets connection
- Inbound OSA-Express QDIO, forwarded outbound over OSA-Express QDIO connections
- Inbound HiperSockets, forwarded outbound over the dynamic XCF HiperSockets connection

No Indicates that this connection is not eligible for the QDIOACCELERATOR function.

Yes Indicates that this connection is eligible for the QDIOACCELERATOR function.

For each entry that represents an affinity created by the TIMEDAFFINITY parameter on the VIPADISTRIBUTE profile statement, displays the preceding information plus the following affinity-related information.

CfgTimAff
The affinity value that was defined in the TIMEDAFFINITY parameter on the VIPADISTRIBUTE profile statement.

TimeAffCnt
The count of currently established connections associated with this affinity.

TimAffLft
The number of seconds left before the affinity between the client IP address and the dynamic VIPA destination IP address and port is removed. After the last established connection is closed, the affinity will remain for the number of seconds indicated in the CfgTimAff field.

Netstat VDPT/-O report

This report displays the dynamic VIPA destination port table information. The command first displays information about distribution to TCP/IP stacks; this section of the report applies to Base targets, tier 1 targets that are not configured to use GRE routing, and tier 2 targets. The command then displays information about distribution to non-z/OS targets; this section of the report applies to tier 1 targets that are configured to use Generic Routing Encapsulation (GRE) routing. The destination port tables exist only on distributing stacks, which are stacks on which a VIPADISTRIBUTE DEFINE keyword was specified.

TSO syntax:

```
►►──NETSTAT VDPT──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────────►◄
```

Modifier:

```
►►──DETAIL─────────────────────────────────────────────────────────────────────►◄
```

DETAIL
Displays the general dynamic VIPA destination port table information plus the Workload Manager weight value and QoS policy action name information as follows:

- The component values of the target server responsiveness (TSR) value
- The count of currently active connections
- The Workload Manager weight value and QoS policy action name information

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

Filter:

```
                ┌──────────────────────────┐
                ▼                          │
►►──┬─IPAddr──────┬─ipaddr─────────────────┴──┬──────────────────────►◄
    │             ├─ipaddr/prefixLen────┤     │
    │             └─ipaddr/subnetmask───┘     │
    │            ┌─────────────────┐          │
    │            ▼                 │          │
    ├─IPPort─────ipaddr+portnum────┴──────────┤
    │        ┌──────────────┐                 │
    │        ▼              │                 │
    └─POrt────────portnum───┴─────────────────┘
```

z/OS UNIX syntax:

```
►►──netstat -O──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ Filter ├──────────►◄
```

Modifier:

```
►►──DETAIL────────────────────────────────────────────────────────────────►◄
```

DETAIL
Displays the general dynamic VIPA destination port table information plus the Workload Manager weight value and QoS policy action name information as follows:

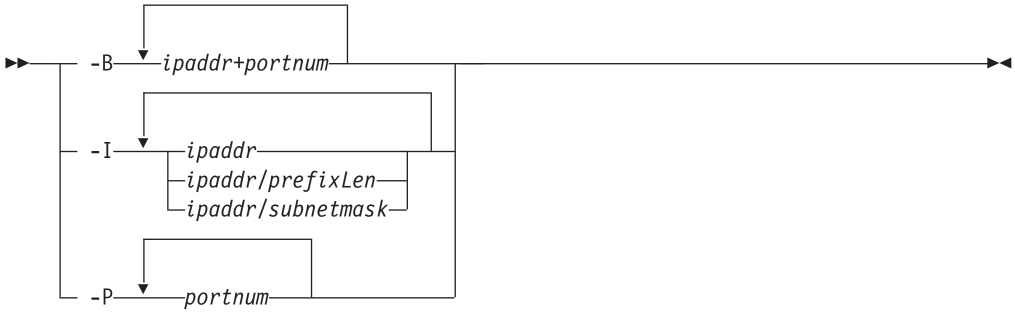
- The component values of the target server responsiveness (TSR) value
- The count of currently active connections
- The Workload Manager weight value and QoS policy action name information

Target:

Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Filter:



Filter description:

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*
Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the VDPT/**-O** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength
Filter the output of the VDPT/**-O** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask
Filter the output of the VDPT/**-O** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be a destination IP address, or a destination XCF IP address.
2. For an IPv6-enabled stack:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.

- An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and will usually provide the same result as its IPv4 address.

Restrictions:

1. The IPAddr/**-I** option for VDPT/**-O** report does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
3. For a UDP endpoint socket, the filter value applies only to the local or source IP address.

IPPort/-B *ipaddr+portnum*

Filter the report output of the VDPT/-O report using the specified IP address and port number. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length, denoting a single IPv4 IP address; each specified IPv6 *ipaddr* value can be up to 45 characters in length, denoting a single IPv6 IP address. Valid *portnum* values are in the range 0 – 65 535. The filter values *ipaddr* and *portnum* will match any combination of the local and remote IP address and local and remote port.

Guidelines:

- The filter value *ipaddr* can be a destination IP address, or a destination XCF IP address.
- For an IPv6-enabled stack, the following apply:
  - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPPort/**-B** option.
  - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as the IPv4 address.
  - For a UDP endpoint socket, the filter value applies only to the local or source IP address and port.

Restrictions:

- The *ipaddr* value in the IPPort/**-B** filter does not support wildcard characters.
- For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
- An entry is returned only when both the *ipaddr* and *portnum* values match.

POrt/-P *portnum*

Filter the output of the VDPT/**-O** report using the specified port number *portnum*. You can enter up to six filter values.

Guideline: The port number can be either a local or remote port.

Restriction: For a UDP endpoint socket, the filter value applies only to the local or source port.

Command syntax examples:

From TSO environment:

```
NETSTAT VDPT
Displays the dynamic VIPA Destination Port Table information in the default TCP/IP
stack.
NETSTAT VDPT TCP TCPCS6
Displays the dynamic VIPA Destination Port Table information in the TCPCS6 stack.
```

From UNIX shell environment:

```
netstat -O
netstat -O -p tcpcs6
```

Report examples:
The following examples are generated using the TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT VDPT

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:35:26
Dynamic VIPA Destination Port Table for TCP/IP Stacks:
Dest IPaddr     DPort DestXCF Addr    Rdy TotalConn  WLM TSR  Flg
-----------     ----- ------------    --- ---------  --- ---  ---
201.2.10.11     00021 201.1.10.15     001 0000310485 01  075  DI
201.2.10.13     00243 201.3.10.16     001 0000256794 03  085
201.2.10.14     00244 201.3.10.16     000 0000000000 15  100  1
201.2.10.15     05000 201.3.10.15     001 0000034011 10  100
201.2.10.18     04040 201.3.10.16     001 0000063421 30  100  2
201.2.10.18     04040 201.3.10.15     001 0000019011 07  100  2
201.4.10.15     07000 201.3.10.16     001 0000094011 10  100  V
201.4.10.15     07000 201.3.10.17     001 0000000000 10  100  K

Dynamic VIPA Destination Port Table for non-z/OS targets:
Dest IPaddr     DPort Target Addr    Rdy TotalConn  Wt  CWt  Flg
-----------     ----- ------------   --- ---------  --- ---  ---
201.2.10.21     03000 205.1.10.15    001 0000310485 01  100  I
201.2.10.22     04011 205.2.10.10    001 0000103162 01  075
201.2.10.22     04011 205.2.10.12    001 0000102658 01  075
201.2.10.25     05000 204.3.10.15    001 0000034011 10  100
```

```
NETSTAT VDPT DETAIL

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:35:26
Dynamic VIPA Distribution Port Table for TCP/IP Stacks:
Dest IPaddr     DPort DestXCF Addr    Rdy TotalConn  WLM TSR  Flg
-----------     ----- ------------    --- ---------  --- ---  ---
201.2.10.11     00021 201.1.10.15     001 0000310485 01  075  DI
  DistMethod: Roundrobin
  TCSR: 100  CER: 075 SEF: 075
  ActConn:    0000000042
201.2.10.13     00243 201.3.10.16     001 0000256794 03  090
  DistMethod: BaseWLM
  TCSR: 100  CER: 095 SEF: 090
  Weight: 12
    Raw          CP: 13 zAAP: 00 zIIP: 10
    Proportional CP: 08 zAAP: 00 zIIP: 04
  ActConn:    0000000091
  QosPlcAct:  *DEFAULT*                                       W/Q: 01
201.2.10.14     00244 201.3.10.16     000 0000000000 15  090  1
  DistMethod: ServerWLM
  TCSR: 100  CER: 095  SEF: 090
  Weight: 60
    Raw          CP: 60 zAAP: 00 zIIP: 60
    Proportional CP: 06 zAAP: 00 zIIP: 54
  Abnorm: 0000         Health: 100
  ActConn:    0000000000
  QosPlcAct:  *DEFAULT*                                       W/Q: 01
201.2.10.15     05000 201.3.10.15     001 0000034011 10  100  A
  DistMethod: WeightedActive
  TCSR: 100  CER: 100  SEF: 100
  Abnorm: 0000         Health: 100
  ActConn:    0000003011
201.2.10.18     04040 201.3.10.16     001 0000063421 30  100  2
  DistMethod: ServerWLM
  TCSR: 100 CER: 100 SEF: 100
  Abnorm: 0000       Health: 100
  ActConn:    0000006006
201.2.10.18     04040 201.3.10.15     001 0000019011 07  100  2
  DistMethod: ServerWLM
  TCSR: 100 CER: 100 SEF: 100
  Abnorm: 0000       Health: 100
  ActConn:    0000003006
201.4.10.15     07000 201.3.10.16     001 0000094011 10  100  V
  DistMethod: HotStandby              SrvType: Preferred
  TCSR: 100  CER: 100  SEF: 100
  Abnorm: 0000         Health: 100
  ActConn:    0000001011
201.4.10.15     07000 201.3.10.17     001 0000000000 10  100  K
  DistMethod: HotStandby              SrvType: Backup
  TCSR: 100  CER: 100  SEF: 100
  Abnorm: 0000         Health: 100
  ActConn:    0000000000
```

```
Dynamic VIPA Destination Port Table for non-z/OS targets:
Dest IPaddr     DPort Target Addr    Rdy TotalConn  Wt  CWt  Flg
-----------     ----- ------------   --- ---------  --- ---  ---
201.2.10.21     03000 205.1.10.15    001 0000310485 01  100  I
  DistMethod: Roundrobin
  T1Wt: 33
  ActConn:    0000000042
201.2.10.22     04011 205.2.10.10    001 0000103162 01  075
  DistMethod: TargCtrl
  T1Wt: 54
  ActConn:    0000000834
201.2.10.22     04011 205.2.10.12    001 0000102658 01  075
  DistMethod: TargCtrl
  T1Wt: 25
  ActConn:    0000000091
201.2.10.25     05000 204.3.10.15    001 0000034011 10  100
  DistMethod: WeightedActive
  T1Wt: 62
  ActConn:    0000001021
```

IPv6 enabled or request for LONG format:

NETSTAT VDPT

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:37:51
Dynamic VIPA Destination Port Table for TCP/IP Stacks:
Dest:        201.2.10.11..21
  DestXCF:   201.1.10.15
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 075
  DistMethod: Roundrobin
  Flg: Dynamic, Inactive
Dest:        201.2.10.13..243
  DestXCF:   201.3.10.16
  TotalConn: 0000000000  Rdy: 001  WLM: 08 TSR: 085
  DistMethod: BaseWLM
  Flg:
Dest:        201.2.10.14..244
  DestXCF:   201.3.10.16
  TotalConn: 0000000000  Rdy: 001  WLM: 15 TSR: 090
  DistMethod: ServerWLM
  Flg: Tier1
Dest:        201.2.10.15..5000
  DestXCF:   201.3.10.15
  TotalConn: 0000034011  Rdy: 001  WLM: 10 TSR: 100
  DistMethod: WeightedActive
  Flg:
Dest:        201.2.10.18..4040
  DestXCF:   201.3.10.16
  TotalConn: 0000063421  Rdy: 001  WLM: 30 TSR: 100
  DistMethod: ServerWLM
  Flg: Tier2
Dest:        201.2.10.18..4040
  DestXCF:   201.3.10.15
  TotalConn: 0000019011  Rdy: 001  WLM: 07 TSR: 100
  DistMethod: ServerWLM
  Flg: Tier2
Dest:        201.4.10.15..7000
  DestXCF:   201.3.10.16
  TotalConn: 0000094011  Rdy: 001  WLM: 10 TSR: 100
  DistMethod: HotStandby           SrvType: Preferred
  Flg: Active
Dest:        201.4.10.15..7000
  DestXCF:   201.3.10.17
  TotalConn: 0000000000  Rdy: 001  WLM: 10 TSR: 100
  DistMethod: HotStandby           SrvType: Backup
  Flg: Backup
DestIntf:
  Dest:        2001:0db8::522:f103..20
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 094
    DistMethod: BaseWLM
    Flg:
DestIntf:
  Dest:        2001:0db8::522:f103..21
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 15 TSR: 100
    DistMethod: ServerWLM
    Flg:
```

```
Dynamic VIPA Destination Port Table for non-z/OS targets:
Dest:        201.2.10.21..3000
  Target Addr: 205.1.10.15
  TotalConn: 0000310485  Rdy: 001  Wt: 01 CWt: 100
  DistMethod: Roundrobin
  Flg: Inactive
Dest:        201.2.10.22..4011
  Target Addr: 205.2.10.10
  TotalConn: 0000103162  Rdy: 001  Wt: 01 CWt: 075
  DistMethod: TargCtrl
  Flg:
Dest:        201.2.10.22..4011
  Target Addr: 205.2.10.12
  TotalConn: 0000102658  Rdy: 001  Wt: 01 CWt: 075
  DistMethod: TargCtrl
  Flg:
Dest:        201.2.10.25..5000
  Target Addr: 204.3.10.15
  TotalConn: 0000034011  Rdy: 001  Wt: 10 CWt: 100
  DistMethod: WeightedActive
  Flg:
DestIntf: INTFNAMB
  Dest: 2001:0db8::522:f222..8000
    Target Addr: 2001:0db8::540:f301
    TotalConn: 0000000000  Rdy: 001  Wt: 14 CWt: 100
    DistMethod: TargCtrl
    Flg:
DestIntf: INTFNAMB
  Dest: 2001:0db8::522:f222..8000
    Target Addr: 2001:0db8::540:f302
    TotalConn: 0000000000  Rdy: 001  Wt: 14 CWt: 075
    DistMethod: TargCtrl
    Flg:
```

NETSTAT VDPT DETAIL

```
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           15:37:51
Dynamic VIPA Destination Port Table for TCP/IP Stacks:
Dest:        201.2.10.11..21
  DestXCF:   201.1.10.15
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 075
  DistMethod: Roundrobin
  Flg: Dynamic, Inactive
  TCSR: 100  CER: 075 SEF: 100
  ActConn:   0000000000
Dest:        201.2.10.13..243
  DestXCF:   201.3.10.16
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 090
  DistMethod: BaseWLM
  Flg:
  TCSR: 100  CER: 095 SEF: 090
  Weight: 12
    Raw          CP: 13 zAAP: 00 zIIP: 10
    Proportional CP: 08 zAAP: 00 zIIP: 04
  ActConn:   0000000000
  QosPlcAct: *DEFAULT*
    W/Q: 01
Dest: 201.2.10.14..244
  DestXCF: 201.3.10.16
  TotalConn: 0000000000 Rdy: 001 WLM: 15 TSR: 090
  DistMethod: ServerWLM
  Flg: Tier1
  TCSR: 100  CER: 095 SEF: 090
  Weight: 60
    Raw          CP: 50 zAAP: 00 zIIP: 61
    Proportional CP: 05 zAAP: 00 zIIP: 55
  Abnorm: 0000        Health: 100
  ActConn:   00000000
  QosPlcAct: *DEFAULT*
    W/Q: 15
Dest: 201.2.10.15..5000
  DestXCF: 201.3.10.15
  TotalConn: 0000034011 Rdy: 001 WLM: 10 TSR: 100
  DistMethod: WeightedActive
  Flg:
  TCSR: 100 CER: 100 SEF: 100
  Abnorm: 0000       Health: 100
  ActConn:      00003011
Dest:        201.2.10.18..4040
  DestXCF:   201.3.10.16
  TotalConn: 0000063421  Rdy: 001  WLM: 15 TSR: 100
  DistMethod: ServerWLM
  Flg: Tier2
  TCSR: 100 CER: 100 SEF: 100
  Weight: 60
    Raw          CP: 60 zAAP: 00 zIIP: 00
    Proportional CP: 60 zAAP: 00 zIIP: 00
  Abnorm: 0000       Health: 100
  ActConn:   00006006
  QosPlcAct: *DEFAULT*
    W/Q: 15
```

```
Dest:        201.2.10.18..4040
  DestXCF:   201.3.10.15
  TotalConn: 0000055421  Rdy: 001  WLM: 07 TSR: 100
  DistMethod: ServerWLM
  Flg: Tier2
  TCSR: 100 CER: 100 SEF: 100
  Weight: 60
    Raw          CP: 60 zAAP: 00 zIIP: 00
    Proportional CP: 60 zAAP: 00 zIIP: 00
  Abnorm: 0000       Health: 100
  ActConn:   0000003006
  QosPlcAct: *DEFAULT*
    W/Q: 07
Dest:        201.4.10.15..7000
  DestXCF:   201.3.10.16
  TotalConn: 0000094011  Rdy: 001  WLM: 10 TSR: 100
  DistMethod: HotStandby           SrvType: Preferred
  Flg: Active
  TCSR: 100  CER: 100  SEF: 100
  Abnorm: 0000         Health: 100
  ActConn:    0000001011
Dest:        201.4.10.15..7000
  DestXCF:   201.3.10.17
  TotalConn: 0000000000  Rdy: 001  WLM: 10 TSR: 100
  DistMethod: HotStandby           SrvType: Backup
  Flg: Backup
  TCSR: 100  CER: 100  SEF: 100
  Abnorm: 0000         Health: 100
  ActConn:    0000000000
DestIntf:
  Dest:        2001:0db8::522:f103..20
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 100
    DistMethod: BaseWLM
    Flg:
    TCSR: 100  CER: 100 SEF: 100
    Weight: 16
      Raw          CP: 24 zAAP: 00 zIIP: 08
      Proportional CP: 12 zAAP: 00 zIIP: 04
    ActConn:   0000000000
    QosPlcAct: *DEFAULT*
      W/Q: 00
DestIntf:
  Dest:        2001:0db8::522:f103..21
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 15 TSR: 100
    DistMethod: ServerWLM
    Flg:
    TCSR: 100  CER: 100 SEF: 100
    Weight: 50
      Raw          CP: 60 zAAP: 00 zIIP: 49
      Proportional CP: 06 zAAP: 00 zIIP: 44
    Abnorm:0000         Health: 100
    ActConn:   0000000000
    QosPlcAct: *DEFAULT*
      W/Q: 15
```

```
Dynamic VIPA Destination Port Table for non-z/OS targets:
Dest:        201.2.10.21..3000
  Target Addr: 205.1.10.15
  TotalConn: 0000310485  Rdy: 001  Wt: 01 CWt: 100
  DistMethod: Roundrobin
  Flg: Inactive
  T1Wt: 33
  ActConn:      00000042
Dest:        201.2.10.22..4011
  Target Addr: 205.2.10.10
  TotalConn: 0000103162  Rdy: 001  Wt: 01 CWt: 075
  DistMethod: TargCtrl
  Flg:
  T1Wt: 54
  ActConn:      00000834
Dest:        201.2.10.22..4011
  Target Addr: 205.2.10.12
  TotalConn: 0000102658  Rdy: 001  Wt: 01 CWt: 075
  DistMethod: TargCtrl
  Flg:
  T1Wt: 25
  ActConn:      00000091
Dest:        201.2.10.25..5000
  Target Addr: 204.3.10.15
  TotalConn: 0000034011  Rdy: 001  Wt: 10 CWt: 100
  DistMethod: WeightedActive
  Flg:
  T1Wt: 62
  ActConn:      00001021
DestIntf:      INTFNAMB
  Dest:        2001:0db8::522:f222..8000
    Target Addr: 2001:0db8::540:f301
    TotalConn: 0000000000  Rdy: 001  Wt: 05 CWt: 100
    DistMethod: TargCtrl
    Flg:
    T1Wt: 500
    ActConn:      00000000
DestIntf:      INTFNAMB
  Dest:        2001:0db8::522:f222..8000
    Target Addr: 2001:0db8::540:f302
    TotalConn: 0000000000  Rdy: 001  Wt: 14 CWt: 200
    DistMethod: TargCtrl
    Flg:
    T1Wt: 700
    ActConn:      00000000
```

Report field descriptions for the Dynamic VIPA Destination Port Table for TCP/IP Stacks:
Displays information about distribution to TCP/IP stacks, including base targets and z/OS tier 1 and tier 2 targets.

For a SHORT format report:

Dest IPaddr
The DVIPA address for which workload is being distributed.

DPort Connections for this port are to be distributed.

DestXCF Addr
The dynamic XCF address of target stack to receive connections.

Flg Flags; depending on the VIPADISTRIBUTE configuration parameters, the state of the target, and the path to the target, flags can have the following values:

1 Indicates that this is a tier 1 DVIPA address.

2 Indicates that this is a tier 2 DVIPA address.

D Indicates a dynamically assigned destination/port entry.

I Indicates that the data path to the target stack is inactive.

K Indicates that this is currently a backup (hot standby) server.

L Indicates that the target stack specified by the DestXCF Addr value is currently processing outbound connections that originated on the target stack for this destination and port pair locally.

V Indicates that this is currently the active server.

For a LONG format report:

DestIntf
The name of this IPv6 interface.

Dest The DVIPA address and port for which workload is being distributed.

DestXCF
The dynamic XCF address of target stack to receive connections.

Flg Flags; depending on the VIPADISTRIBUTE configuration parameters, the state of the target, and the path to the target, flags can have the following values:

Active Indicates that this is currently the active server.

Backup
Indicates that this is currently a backup (hot standby) server.

Dynamic
Indicates a dynamically assigned destination/port entry.

Inactive
Indicates that the datapath to the XCF target is inactive.

Local Indicates that the target stack specified by the DestXCF Addr value is currently processing outbound connections for this destination and port pair locally.

Tier1 Indicates that this is a tier 1 DVIPA address.

Tier2 Indicates that this is a tier 2 DVIPA address.

For a SHORT or LONG format report:

DistMethod
The distribution method in use.

BaseWLM
Indicates that WLM system weights and policy information are used to distribute incoming connection requests.

HotStandby
Indicates that HotStandby distribution is in use.

Roundrobin
Indicates that incoming connection requests are distributed using the round-robin method.

ServerWLM
Indicates that WLM server weights and policy information are used to distribute incoming connection requests.

WLM server weights are used if SERVERWLM was specified on the VIPADISTRIBUTE statement for this DVIPA and port, and all target servers are able to provide WLM server-specific weights. Otherwise, BaseWLM is used.

WeightedActive
Indicates that incoming connection requests are distributed using the weighted active connections method.

Rule: For a short format report, you need to use the DETAIL modifier to display this field.

Rdy The number of applications ready to receive connections. A count of 0 indicates that there are no applications on this target stack ready to receive new connections. Either the application has not been started on this target, or if it was started, it might have been terminated or quieseced with the `Vary TCPIP,,SYSPLEX,QUIesce` command.

SrvType
Indicates the server type when the HotStandby distribution method is configured.

Backup
Indicates that this is a backup target. A backup target is initially a hot standby target. Connections are not distributed to hot standby targets. If the active target becomes unavailable, the distributor switches targets and one of the hot standby targets becomes the active target.

Preferred
Indicates that this is the preferred target. If AUTOSWITCHBACK is configured, then the preferred target is the active target if it is available and has not had any health problems. If the active target becomes unavailable, the distributor switches to use a hot standby target; the active target becomes a hot standby target and the selected hot standby target becomes the active target.

Rule: For a short format report, you need to use the DETAIL modifier to display this field.

TotalConn
The total number of connections that have been forwarded to the stack identified by DestXCF Addr. This field will wrap.

WLM When the distribution method is BASEWLM or SERVERWLM this is the Workload Manager weight value for the target listener. The weight value is either an indication of the target system's capacity for additional work or the more granular indication of the specific server's capacity for additional work, based on how well it is meeting its WLM policy goals (where higher numbers indicate a server with greater capacity). WLM system weights are indicated by the BaseWLM flag. WLM server-specific weights are indicated by the ServerWLM flag.

When this is a tier 1 target, the original system weight or server-specific weight might be modified by adding the weight of a tier 2 target server for the same group name that is also on this TCP/IP stack, therefore, the displayed value indicates the comparative fitness of a tier 1 target in terms of both the tier 1 and the corresponding tier 2 server capacity for additional work. The weights represent normalized weights; the original weights of the tier 1 and tier 2 server (if any) are added together and proportionally reduced for use by the distribution algorithm. Connections are distributed to these servers in a weighted distribution using the normalized weights.

The weights represent normalized weights; the original raw weights received from WLM are modified by multiplying them by the target server responsiveness (TSR) value and are proportionally reduced for use by the distribution algorithm. Thus, the displayed value indicates the comparative fitness of a server both in terms of system or server capacity and in terms of TCP connection setup responsiveness. Connections are distributed to these servers in a weighted round-robin manner using the normalized weights.

For more information about WLM, see Sysplex distributor details in the z/OS Communications Server: IP Configuration Guide.

When the distribution method is WEIGHTEDACTIVE, this value is the configured weight for the target listener. This weight is used by the distributor to determine the proportion of incoming requests to route to this target such that the number of active connections on each target is proportionally equivalent to the configured weight for each target.

TSR The target server responsiveness value for the target server.

The sysplex distributor monitors the ability of a target server to process new connections. At each interval of approximately one minute, it generates a target server responsiveness fraction percentage to indicate how well the server is accepting new TCP connection setup requests. It is not a measure of how well the server is servicing the connections.

- The value 100 indicates that the target server is successfully accepting all new TCP connection setup requests. A value of 100 is also displayed for target stacks at a pre-V1R7 z/OS level. If there is at least one target stack for this DVIPA and a port that is at a pre-V1R7 z/OS level, then no target server responsiveness calculations are applied to the WLM values.
- A value that is greater than 0 but less than 100 indicates that the server is having problems accepting some new connection requests. These problems can be due to network connectivity, server application problems, or target stack problems.
- The value 0 indicates that the target server is unable to process new connection requests. This can be due to network connectivity, server application problems, or target stack problems. No new TCP connection setup requests are distributed to a target server with a TSR value of 0.

The sysplex distributor modifies the WLM weight for each target server by the calculated target server responsiveness percentage, and, after normalizing the weights, uses these new values to weight its distribution of new connection requests to the target servers. For example, if there are three target servers for a particular DVIPA with calculated TSRs of 75, 50, and 100 percent respectively, and, after applying the TSRs to the WLM weights, the normalized weights are 7, 2, and 3, the sysplex distributor would be expected to distribute three and a half times as many new connection requests to the first target server as to the second server and one and half times as many new connection requests to the third server as to the second server.

The TSR percentage is calculated from two component values, the target connectivity success rate (TCSR) and the server accept efficiency fraction (SEF)

- The TCSR measures the percentage of connection setup requests routed from the distributor that are successfully received by the target for this server.
- The SEF measures the effectiveness of the server application in accepting new connection requests and managing its backlog queue.

The values of each of the components are displayed when DETAIL is specified on the command.

DETAIL

Invoking VDPT/**-O** DETAIL displays the VDPT information stated above and includes the following additional information:

TCSR The target connectivity success rate (TCSR) is a measure of the percentage of connection setup requests routed from the distributor that are successfully received by the target for this server. It is displayed as a percentage. A value of 100 indicates that all connection setup requests routed to the target stack destined for this server are being successfully received by the target. A value of 0 indicates that no connection setup requests for this server are successfully reaching the target. This value is one component part of the target server responsiveness (TSR) value.

CER The connection establishment rate (CER) is a measure of the percentage of the connection setup requests received at the target that achieve completion with the client (that is, arrive at connection established state). It is displayed as a percentage. The value 100 indicates that all new connection setup requests are resulting in established connections. The value 0 indicates that no new connection setup requests have become successfully established. This value is used for diagnosis only and is not integrated into the TSR calculation. For information about diagnosing sysplex problems, see the steps for diagnosing sysplex problems in the z/OS Communications Server: IP Diagnosis Guide.

SEF The server accept efficiency fraction (SEF) is a measure, calculated at intervals of approximately one minute, of the efficiency of the server application in accepting new connection requests and managing its backlog queue. It is displayed as a percentage. A value of 100 indicates that the server application is successfully accepting all its new connection setup requests. A value of 0 indicates the server application is not responding to new connection setup requests. This value is one component part of the target server responsiveness (TSR) value.

Weight

The composite weight. This is the sum of the displayed modified CP, zAAP, and zIIP weights that follow.

CP When the distribution method is BASEWLM the following apply:

- The Raw value is the WLM system general CP weight recommendation. It is based on the amount of displaceable general CPU capacity on this system as compared to the other target systems.
- The Proportional value is the Raw value modified by the expected general CP utilization proportion configured on the VIPADISTRIBUTE PROCTYPE statement for this application.

When the distribution method is SERVERWLM the following apply:

- The Raw value is the WLM server-specific general CP recommendation. This is the amount of displaceable general CPU capacity based on the application workload's importance (as defined by the WLM policy) as compared to the other target systems.
- The Proportional value is the Raw value modified by the proportion of general CP capacity that is currently being consumed by the application's workload as compared to the other processors (zAAP and zIIP)

zAAP When the distribution method is BASEWLM the following apply:

- The Raw value is the WLM system zAAP weight recommendation. It is based on the amount of displaceable zAAP capacity on this system as compared to the other target systems.
- The Proportional value is the Raw value modified by the expected zAAP utilization proportion configured on the VIPADISTRIBUTE PROCTYPE statement for this application.

When the distribution method is SERVERWLM the following apply:

- The Raw value is the WLM server-specific zAAP recommendation, which is the amount of displaceable zAAP capacity based on the application workload's importance (as defined by the WLM policy) as compared to the other target systems.
- The Proportional value is the Raw value modified by the proportion of zAAP capacity that is currently being consumed by the application's workload as compared to the other processors (general CPU and zIIP)

zIIP When the distribution method is BASEWLM the following apply:

- The Raw value is the WLM system zIIP weight recommendation. It is based on the amount of displaceable zIIP capacity on this system as compared to the other target systems.
- The Proportional value is the Raw value modified by the expected zIIP utilization proportion configured on the VIPADISTRIBUTE PROCTYPE statement for this application.

When the distribution method is SERVERWLM the following apply:

- The Raw value is the WLM server-specific zIIP recommendation. This is the amount of displaceable zIIP capacity based on the application workload's importance (as defined by the WLM policy) as compared to the other target systems.
- The Proportional value is the Raw value modified by the proportion of zIIP capacity that is currently being

consumed by the application's workload as compared to the other processors (general CPU and zAAP)

ActConn

Indicates the current number of active connections for a target TCP/IP. This count is incremented by the distributing TCP/IP when one of the following occurs:

- A connection request is forwarded by the distributing TCP/IP to that target.
- The distributing TCP/IP is informed that the target has initiated a TCP connection using this DVIPA as a source IP address.

Abnorm

Indicates whether the server application is experiencing conditions under which transactions are completing abnormally. It represents a rate of abnormal transaction completions per 1000 total transaction completions. It is applicable only for TCP applications that act as Subsystem Work Managers and report transaction status using Workload Management Services, such as IWMRPT. For example, a value of 100 indicates that 10% of all transactions processed by the server application are completing abnormally. Under normal conditions, this value is 0. A nonzero value indicates that the server application has reported some abnormal transactions completions to WLM and that WLM has reduced the recommendation provided to sysplex distributor for this server instance. This reduction in the WLM recommendation enables fewer new TCP connections to be directed to servers that are not experiencing problem conditions that result in abnormal transaction completions. The greater the value in the Abnorm field, the greater the reduction WLM applies to the recommendation for this target instance. For more information about the conditions that cause abnormal transaction completions for a given server application, see the documentation that was provided by the server application.

If the distribution method is not SERVERWLM, then this field is 0. For more information about workload management interfaces, see z/OS MVS Programming: Workload Management Services.

Health

The health indicator of the server application. This health indicator is available only for applications that provide this information to WLM using the IWM4HLTH or IWMSRSRG services. It provides a general health indication for an application or subsystem. Under normal circumstances, the value of this field is 100, which indicates that the server is 100% healthy. Any value less than 100 indicates that the server is experiencing problem conditions that are not enabling it to process new work requests successfully. A value of less than 100 also causes the WLM to reduce the recommendation provided to sysplex distributor for this server instance. This reduction in the WLM recommendation enables fewer new TCP connections to be directed to servers that are not experiencing problem conditions. The reduction in the WLM recommendation is proportional to the Health indicator value. For example, given a health value of only 20%, WLM reduces the recommendation for this server by 80%. For more information regarding the conditions that lead to a health indicator of less than 100, see the documentation for the server application.

If applications do not provide this health indicator to WLM, then this field has a value of 100. If the distribution method is not SERVERWLM, then this field has a value of 100. For more information about workload management interfaces, see z/OS MVS Programming: Workload Management Services.

W/Q The Workload Manager weight value for the target server after modification using QoS information provided by the Policy Agent. QoS information is an indication of the following information:

- Network performance (TCP retransmissions and timeouts)
- Maximum connections allowed versus actual connections
- Expected overall throughput versus the actual throughput achieved

This value is used by a distributing stack to determine the quantity of connections to be forwarded to this target stack, relative to other target stacks. Note that if, for a particular incoming connection, a target server's W/Q value for the destination address, port, and QoS policy action is 0, while the W/Q value for the destination address, port, and QoS policy action on other target servers is nonzero, no connections are forwarded to the target server with a 0 W/Q value.

Note: If all target servers for the destination address, port, and QoS policy action have 0 W/Q values, connection forwarding is done in a round-robin fashion, rather than based on WLM or QoS information.

QosPlcAct
The QoS policy action name configured to the Policy Agent. If multiple QoS policy actions are configured for a single destination address and port, each policy action name is displayed along with its associated WLM and W/Q values. A QosPolicyAction of *Default* indicates the WLM and W/Q values used when there is no QosPolicyAction that applies to an incoming connection.

Report field descriptions for the Dynamic VIPA Destination Port Table for non-z/OS targets:
Displays information about distribution to non-z/OS targets. This includes tier 1 targets.

For a SHORT format report:

Dest IPaddr
The DVIPA address for which workload is being distributed.

DPort Connections for this port that are to be distributed.

Target Addr
The IP Address of a non-z/OS target that is to receive connections. This is a tier 1 target (for example, a DataPower appliance).

Flg Flags; can have one of the following values:

A Indicates that incoming connection requests are distributed by the weighted active connections method.

I Indicates that the data path to the target is inactive.

R Indicates that incoming connection requests are distributed by the round-robin method.

T Indicates that incoming connection requests are distributed using weights that are provided by the tier 1 targets.

For a LONG format report:

DestIntf
The name of this IPv6 interface.

Dest The DVIPA address and port for which workload is being distributed.

Target Addr
The IP address of a non-z/OS target that is to receive connections. This is a tier 1 target (for example, a DataPower appliance).

Flg Flags; can have one of the following values:

Inactive
Indicates that the datapath to the target is inactive.

Roundrobin
Indicates that incoming connection requests are distributed by the round-robin method.

TargetCtrl
Indicates that incoming connection requests are distributed using weights provided by the tier 1 targets.

WeightedActive
Indicates that incoming connection requests are distributed by the weighted active connections method.

For a SHORT or LONG format report:

Rdy This field indicates whether there are applications ready to receive connections. The value 0 indicates that there are no application on this target that is ready to receive new connections. The value 1 indicates that there is an application on this target that is ready to receive new connections.

TotalConn
The total number of connections that have been forwarded to the target identified by Target Addr. This field will wrap.

Wt The composite weight of the target. The following distribution methods are supported:

TARGCONTROLLED
The weights represent normalized weights; the original weights received from the tier 1 targets (if any) might be modified by multiplying them by the central processor complex (CPC) weight (CWt) value and are proportionally reduced for use by the distribution algorithm. The displayed value indicates the comparative fitness of a target both in terms of target capacity and in terms of the CPC availability of the associated tier 2 target servers. Connections are distributed to these servers in a weighted distribution using the normalized weights.

WEIGHTEDACTIVE

The CPC weight 0 affects weighted active distribution. If at least one nonzero CPC weight has been received for a group of associated tier 2 target servers, then the following occurs:

- If the associated CPC weight is 0, then the normalized weight is set to 0 for this target.
- If the CPC weight is not 0, then normalization uses the weight that is configured.

ROUNDROBIN

The CPC weight 0 affects round-robin distribution. If at least one nonzero CPC weight has been received for a group of associated tier 2 target servers, then the following occurs:

- If the associated CPC weight is 0, then the normalized weight is set to 0 for this target.
- If the CPC weight is not 0, the normalized weight for this target is set to 1.

CWt A value that represents the combined weights of the tier 2 target servers on the stacks that are on the same central processor complex (CPC) as the tier 1 target. These applications are the tier 2 targets of connection requests from the tier 1 target. This value can be used to modify the weight value that is provided by the tier 1 target. See z/OS Communications Server: IP Configuration Guide for more details.

DETAIL

Details include the following fields:

T1Wt The weight that is received from the tier 1 target.

ActConn

Indicates the current number of active connections for this target. This count is incremented by the distributing TCP/IP when a connection request is forwarded by the distributing TCP/IP to that target.

Netstat VIPADCFG/-F report

Displays the dynamic VIPA configuration for a local host.

TSO syntax:

```
►►──NETSTAT VIPADCFG──┤ Modifier ├──┤ Target ├──┤ Output ├──┤ (Filter ├──────────►◄
```

Modifier:

```
►►──DETAIL──────────────────────────────────────────────────────────────────►◄
```

DETAIL

Displays the general dynamic VIPA configuration information, along with the following information:

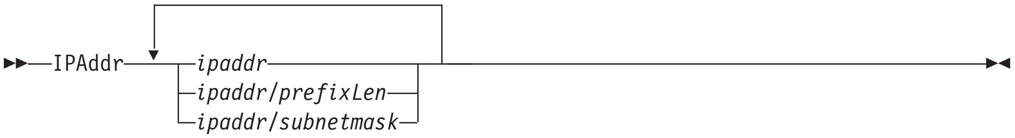
- The OPTLOCAL value.
- If the distribution method is WeightedActive, displays the configured active connection weight.
- If the distribution method is BASEWLM, displays the PROCTYPE parameters.

- If the distribution method is SERVERWLM, diplays the PROCXCOST and ILWEIGHTING parameters.

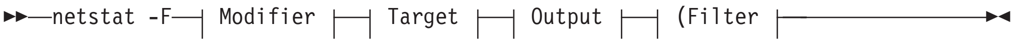
Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see “The TSO NETSTAT command syntax” on page 306 or “Netstat command output” on page 316.

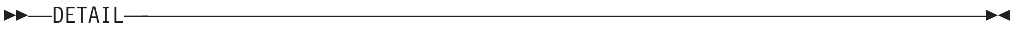
Filter:



z/OS UNIX syntax:



Modifier:

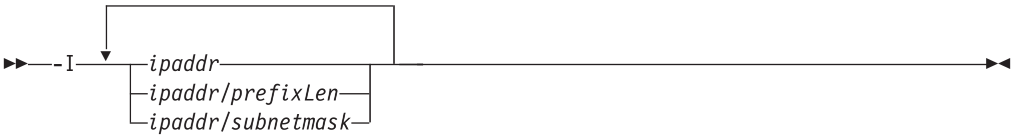


DETAIL
Displays the general dynamic VIPA configuration information and the OPTLOCAL value.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See “The Netstat command target” on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see “The z/OS UNIX netstat command syntax” on page 310 or “Netstat command output” on page 316.

Filter:



Filter description:

IPAddr/-I *ipaddr***IPAddr/-I** *ipaddr/prefixlength***IPAddr/-I** *ipaddr/subnetmask*

Filter the report output using the specified IP address *ipaddr*, *ipaddr/prefixlength*, or *ipaddr/subnetmask*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters in length and each selected IPv6 *ipaddr* value can be up to 45 characters in length.

ipaddr Filter the output of the VIPADCFG/**-F** report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default *prefixlength* of 128 is used.

ipaddr/prefixlength

Filter the output of the VIPADCFG/**-F** report using the specified IP address and prefix length *ipaddr/prefixlength*. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

ipaddr/subnetmask

Filter the output of the VIPADCFG/**-F** report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be an IPv4 IP address.

Guidelines:

1. The filter value *ipaddr* can be a dynamic VIPA address, a destination IP address, or a destination XCF IP address.
2. For an IPv6-enabled stack the following apply:
   - Both IPv4 and IPv6 *ipaddr* values are accepted and can be mixed on the IPAddr/**-I** option.
   - An IPv4-mapped IPv6 address is accepted as a valid *ipaddr* value and usually provides the same result as its IPv4 address.

Restrictions:

1. The IPAddr/**-I** option for the VIPADCFG/**-F** report does not support wildcard characters.
2. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.

Command syntax examples:

From TSO environment:

```
NETSTAT VIPADCFG
   Display the dynamic VIPA configuration for the default TCP/IP stack.
NETSTAT VIPADCFG TCP TCPCS6
   Display the dynamic VIPA configuration for the TCPCS6 stack.
```

From UNIX shell environment:

```
   netstat -F
   netstat -F -p tcpcs6
```

Report examples:

The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

If the TCP/IP stack is not currently in the sysplex group, two messages will preceed the report, one indicating that the TCP/IP stack is not a member of the sysplex group and the other indicating that all dynamic VIPA configuration for the TCP/IP stack is currently inactive. See z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM), messages EZZ2502I and EZZ2503I respectively, for detailed information about these messages.

If the stack is delaying sysplex profile processing because VTAM or OMPROUTE is not initialized, VIPADYNAMIC configuration information is not available and message EZZ2505I precedes the report heading. See z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM) for more information.

Not IPv6 enabled (SHORT format):

```
NETSTAT VIPADCFG
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS          19:47:49
Dynamic VIPA Information:

 VIPA Backup:
   IP Address       Rank    Address Mask     Moveable  SrvMgr Flg
   ----------       ----    ------------     --------  ------ ---
   201.2.10.29      025     255.255.255.192  WhenIdle   Yes
   201.2.10.30      100     255.255.255.192  Immediate No
   201.2.10.32      040
   201.2.10.40      010     255.255.255.192  Immediate No      1
   201.2.10.45      020     255.255.255.192  Immediate No      2
   201.2.10.54      010     255.255.255.192  Immediate No      C

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr Flg
   ----------       -----------      --------  ------ ---
   201.2.10.11      255.255.255.192  WhenIdle  No
   201.2.10.12      255.255.255.192  Immediate Yes
   201.2.10.13      255.255.255.192  Immediate No
   201.2.10.14      255.255.255.192  Immediate No
   201.2.10.17      255.255.255.192  Immediate No      1
   201.2.10.18      255.255.255.192  Immediate No      2C
   201.2.10.19      255.255.255.192  Immediate No      C
   201.2.10.31      255.255.255.192  Immediate No      1

 VIPA Range:
   AddressMask      IP Address       Moveable  SAF Name
   -----------      ----------       --------  --------
   255.255.255.192  201.2.10.192     NonDisr   RANGE1
   255.255.255.192  201.2.20.192     Disrupt

 VIPA Distribute:
   IP Address       Port   XCF Address      SysPt  TimAff  Flg
   ----------       ----   -----------      -----  ------  ----
   201.2.10.11      n/a    ALL              Yes    200
   201.2.10.13      243    ALL              No     No      O
   201.2.10.14      244    ALL              No     No      1
   201.2.10.15      5000   201.3.10.15      No     No      A
   201.2.10.17      8080   200.1.10.10      No     Yes     1
   201.2.10.18      4040   201.3.10.16      No     Yes     2
   201.2.10.18      4040   201.3.10.15      No     Yes     2
   201.4.10.15      7000   201.3.10.16      No     No
   201.4.10.15      7000   201.3.10.17      No     No

VIPA Service Manager:
   McastGroup: 224.0.0.1        Port: 04444  Pwd: Yes

 VIPA Route:
   XCF Address     TargetIp
   -----------     --------
   201.10.10.1     201.20.20.1
   201.10.10.2     201.20.20.2
   201.10.10.3     201.20.20.3

Deactivated Dynamic VIPA Information:

 VIPA Backup:
   IP Address       Rank    Address Mask     Moveable  SrvMgr Flg
   ----------       ----    ------------     --------  ------ ---
   201.2.10.40      100     255.255.255.192  Immediate No

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr Flg
   ----------       -----------      --------  ------ ---
   201.2.10.20      255.255.255.192  Immediate No

 VIPA Distribute:
   IP Address       Port   XCF Address      SysPt  TimAff  Flg
   ----------       ----   -----------      -----  ------  ----
   201.2.10.20      5000    ALL              No     No      B
```

```
NETSTAT VIPADCFG DETAIL
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           19:47:49
Dynamic VIPA Information:

 VIPA Backup:
   IP Address       Rank    Address Mask     Moveable  SrvMgr Flg
   ----------       ----    ------------     --------  ------ ---
   201.2.10.29      025     255.255.255.192  WhenIdle  Yes
   201.2.10.30      100     255.255.255.192  Immediate No
   201.2.10.32      040
   201.2.10.40      010     255.255.255.192  Immediate No     1
   201.2.10.45      020     255.255.255.192  Immediate No     2
   201.2.10.54      010     255.255.255.192  Immediate No     C

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr Flg
   ----------       -----------      --------  ------ ---
   201.2.10.11      255.255.255.192  WhenIdle  No
   201.2.10.12      255.255.255.192  Immediate Yes
   201.2.10.13      255.255.255.192  Immediate No
   201.2.10.17      255.255.255.192  Immediate No     1
   201.2.10.18      255.255.255.192  Immediate No     2C
   201.2.10.19      255.255.255.192  Immediate No     C
   201.2.10.31      255.255.255.192  Immediate No     1

 VIPA Range:
   AddressMask      IP Address       Moveable  SAF Name
   -----------      ----------       --------  --------
   255.255.255.192  201.2.10.192     NonDisr   RANGE1
   255.255.255.192  201.2.20.192     Disrupt

  VIPA Distribute:
   IP Address       Port   XCF Address      SysPt  TimAff  Flg
   ----------       ----   -----------      -----  ------  ----
   201.2.10.11      n/a    ALL              Yes    200     R
     DistMethod: Roundrobin
     OptLoc: No
   201.2.10.13      243    ALL              No     No      0
     DistMethod: BaseWLM
     OptLoc: 1
     ProcType:
       CP: 60  zAAP: 00  zIIP: 40
   201.2.10.14      243    ALL              No     No      1
     DistMethod: ServerWLM
     OptLoc: No
     ProcXCost:
       zAAP: 003  zIIP: 001
     ILWeighting: 1
   201.2.10.17      8080   200.1.10.10      No     Yes     1
     DistMethod: TargCtrl
     OptLoc: No
     GrpName: CICSGROUP       RtgType: GRE CtrlPort: 1010
   201.2.10.18      4040   201.3.10.16      No     Yes     2
     DistMethod: ServerWLM
     OptLoc: No
     GrpName: CICSGROUP
     ProcXCost:
       zAAP: 003  zIIP: 001
     ILWeighting: 1
   201.2.10.18      4040   201.3.10.15      No     Yes     2
     DistMethod: ServerWLM
     OptLoc: No
     GrpName: FTPGROUP
   201.4.10.15      7000   201.3.10.16      No     No
     DistMethod: HotStandby      SrvType: Preferred
     AutoSwitchBack: Yes         HealthSwitch: Yes
     OptLoc: No
   201.4.10.15      7000   201.3.10.17      No     No
     DistMethod: HotStandby      SrvType: Backup  Rank: 001
     AutoSwitchBack: Yes         HealthSwitch: Yes
     OptLoc: No
 VIPA Service Manager:
   McastGroup: 224.0.0.1        Port: 04444  Pwd: Yes

 VIPA Route:
   XCF Address     TargetIp
   -----------     --------
   201.10.10.1     201.20.20.1
   201.10.10.2     201.20.20.2
   201.10.10.3     201.20.20.3

Deactivated Dynamic VIPA Information:

 VIPA Backup:
   IP Address       Rank    Address Mask     Moveable  SrvMgr Flg
   ----------       ----    ------------     --------  ------ ---
   201.2.10.40      100     255.255.255.192  Immediate No

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr Flg
   ----------       -----------      --------  ------ ---
   201.2.10.20      255.255.255.192  Immediate No

 VIPA Distribute:
   IP Address       Port   XCF Address      SysPt  TimAff  Flg
   ----------       ----   -----------      -----  ------  ----
   201.2.10.20      5000    ALL              No     No      B
```

IPv6 enabled or request for LONG format:

```
NETSTAT VIPADCFG
MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           19:49:12
Dynamic VIPA Information:

  VIPA Backup:
    IpAddr/PrefixLen: 201.2.10.29/26
      Rank: 025  Moveable: WhenIdle   SrvMgr: Yes Flg:
    IpAddr/PrefixLen: 201.2.10.30/26
      Rank: 025  Moveable: Immediate  SrvMgr: No  Flg:
    IpAddr/PrefixLen: 201.2.10.32
      Rank: 040  Moveable:            SrvMgr:     Flg:
    IpAddr/PrefixLen: 201.2.10.40
      Rank: 010  Moveable: Immediate  SrvMgr: No  Flg: 1
    IpAddr/PrefixLen: 201.2.10.45
      Rank: 020  Moveable: Immediate  SrvMgr: No  Flg: 2
    IpAddr/PrefixLen: 201.2.10.54
      Rank: 010  Moveable: Immediate  SrvMgr: No  Flg: C
    IntfName: INTFNAM5
      IpAddr: 2001:db8::526:f604/64
        Rank: 050  Moveable: Immediate  SrvMgr: n/a Flg: 2C
    IntfName: INTFNAM4
      IpAddr: 2001:db8::526:f606/64
        Rank: 025  Moveable: WhenIdle   SrvMgr: n/a Flg: C
    IntfName: INTFNAM6
      IpAddr: 2001:db8::526:f603
        Rank: 050  Moveable:            SrvMgr: n/a Flg:

  VIPA Define:
    IpAddr/PrefixLen: 201.2.10.11/26
      Moveable: WhenIdle   SrvMgr: No  Flg:
    IpAddr/PrefixLen: 201.2.10.12/26
      Moveable: Immediate  SrvMgr: Yes Flg:
    IpAddr/PrefixLen: 201.2.10.13/26
      Moveable: Immediate  SrvMgr: No  Flg:
    IpAddr/PrefixLen: 201.2.10.14/26
      Moveable: Immediate SrvMgr: No   Flg:
    IpAddr/PrefixLen: 201.2.10.17/26
      Moveable: Immediate SrvMgr: No  Flg: 1
    IpAddr/PrefixLen: 201.2.10.18/26
      Moveable: Immediate SrvMgr: No  Flg: 2C
    IpAddr/PrefixLen: 201.2.10.19/26
      Moveable: Immediate SrvMgr: No  Flg: C
    IpAddr/PrefixLen: 201.2.10.31/26
      Moveable: Immediate SrvMgr: No Flg: 1
    IntfName: INTFNAM1
      IpAddr: 2001:0db8::522:f103
        Moveable: Immediate  SrvMgr: n/a Flg:
    IntfName: INTFNAM2
      IpAddr: 2001:0db8::522:f203
        Moveable: Immediate  SrvMgr: n/a Flg:
    IntfName: INTFNAMB
      IpAddr: 2001:0db8::522:f222/64
        Moveable: Immediate  SrvMgr: No  Flg: 1
    IntfName: INTFNAMC
      IpAddr: 2001:0db8::522:f333/64
        Moveable: Immediate  SrvMgr: No  Flg: 2C
    IntfName:  INTFNAMD
      IpAddr: 2001:0db8::22:f334/64
        Moveable: Immediate  SrvMgr: No  Flg: C

  VIPA Range:
    IpAddr/PrefixLen: 201.2.10.192/26
      Moveable: NonDisr   SAFName: RANGE1
    IpAddr/PrefixLen: 201.2.20.192/26
      Moveable: Disrupt
    IntfName: INTFNAM3
      IpAddr/PrefixLen: 2001:0db8::522:f303/24
        Moveable: NonDisr   SAFName: RANGE2
```

```
VIPA Distribute:
  Dest:        201.2.10.11..n/a
    DestXCF:   ALL
    DistMethod: Roundrobin
    SysPt:   Yes  TimAff: 200  Flg: Roundrobin
  Dest:        201.2.10.13..243
    DestXCF:   ALL
    DistMethod: BaseWLM
    SysPt:   No   TimAff: No   Flg: OptLocal
  Dest:        201.2.10.14..244
    DestXCF:   ALL
    DistMethod: ServerWLM
    SysPt:   No   TimAff: No   Flg: Tier1
  Dest:        201.2.10.17..8080
    DestXCF:   200.1.10.10
    SysPt:   No   TimAff: Yes  Flg: Tier1
  Dest:        201.2.10.18..4040
    DestXCF:   201.3.10.15
    DistMethod: ServerWLM
    SysPt:   No   TimAff: Yes  Flg: Tier2
  Dest:        201.2.10.18..4040
    DestXCF:   201.3.10.16
    DistMethod: ServerWLM
    SysPt:   No   TimAff: Yes  Flg: Tier2
  Dest:        201.4.10.15..7000
    DestXCF:   201.3.10.16
    DistMethod: HotStandby       SrvType: Preferred
    SysPt:   No   TimAff: Yes  Flg:
  Dest:        201.4.10.15..7000
    DestXCF:   201.3.10.17
    DistMethod: HotStandby       SrvType: Backup  Rank: 001
    SysPt:   No   TimAff: Yes  Flg:
  DestIntf:    INTFNAM1
    Dest:      2001:0db8::522:f103..20
      DestXCF: ALL
      DistMethod: ServerWLM
      SysPt: No   TimAff: No    Flg:
  DestIntf:    INTFNAM1
    Dest:      2001:0db8::522:f103..21
      DestXCF: ALL
      DistMethod: ServerWLM
      SysPt: Yes  TimAff: 10    Flg:
  DestIntf:    INTFNAMB
    Dest:      2001:0db8::522:f222..8000
      DestXCF: 2001:0db8::540:f301
      DistMethod: TargCtrl
      SysPt:   No   TimAff: No   Flg: Tier1
  DestIntf:    INTFNAMB
    Dest:      2001:0db8::522:f222..8000
      DestXCF: 2001:0db8::540:f302
      DistMethod: TargCtrl
      SysPt:   No   TimAff: No   Flg: Tier1

VIPA Service Manager:
  McastGroup: 224.0.0.1
  Port: 04444  Pwd: Yes
```

```
VIPA Route:
    DestXCF:     201.10.10.1
      TargetIp:  201.20.20.1
    DestXCF:     201.10.10.2
      TargetIp:  201.20.20.2
    DestXCF:     2eco::500:f103
      TargetIp:  2eco::100:f103

Deactivated Dynamic VIPA Information:
VIPA Backup:
    IpAddr/PrefixLen: 201.2.10.40/26
      Rank: 025  Moveable: Immediate  SrvMgr: No  Flg:

  VIPA Define:
     IpAddr/PrefixLen: 201.2.10.20/26
      Moveable: Immediate  SrvMgr: No  Flg:

  VIPA Distribute:
    Dest:        201.2.10.20..5000
      DestXCF:   ALL
        SysPt:   No   TimAff: No   Flg: BaseWLM
```

```
NETSTAT VIPADCFG DETAIL
MVS TCP/IP NETSTAT CS V2R1      TCPIP Name: TCPCS           19:49:12
Dynamic VIPA Information:

  VIPA Backup:
    IpAddr/PrefixLen: 201.2.10.29/26
      Rank: 025  Moveable: WhenIdle   SrvMgr: Yes Flg:
    IpAddr/PrefixLen: 201.2.10.30/26
      Rank: 025  Moveable: Immediate  SrvMgr: No  Flg:
    IpAddr/PrefixLen: 201.2.10.32
      Rank: 040  Moveable:            SrvMgr:     Flg:
    IpAddr/PrefixLen: 201.2.10.40
      Rank: 010  Moveable: Immediate  SrvMgr: No  Flg: 1
    IpAddr/PrefixLen: 201.2.10.45
      Rank: 020  Moveable: Immediate  SrvMgr: No  Flg: 2
    IpAddr/PrefixLen: 201.2.10.54
      Rank: 010  Moveable: Immediate  SrvMgr: No  Flg: C
    IntfName: INTFNAM5
      IpAddr: 2001:db8::526:f604/64
        Rank: 050  Moveable: Immediate  SrvMgr: n/a Flg: 2C
    IntfName: INTFNAM4
      IpAddr: 2001:db8::526:f606/64
        Rank: 025  Moveable: WhenIdle   SrvMgr: n/a Flg: C
    IntfName: INTFNAM6
      IpAddr: 2001:db8::526:f603
        Rank: 050  Moveable:            SrvMgr: n/a Flg:

  VIPA Define:
    IpAddr/PrefixLen: 201.2.10.11/26
      Moveable: WhenIdle   SrvMgr: No  Flg:
    IpAddr/PrefixLen: 201.2.10.12/26
      Moveable: Immediate  SrvMgr: Yes Flg:
    IpAddr/PrefixLen: 201.2.10.13/26
      Moveable: Immediate  SrvMgr: No  Flg:
    IpAddr/PrefixLen: 201.2.10.17/26
      Moveable: Immediate SrvMgr: No  Flg: 1
    IpAddr/PrefixLen: 201.2.10.18/26
      Moveable: Immediate SrvMgr: No  Flg: 2C
    IpAddr/PrefixLen: 201.2.10.19/26
      Moveable: Immediate SrvMgr: No  Flg: C
    IpAddr/PrefixLen: 201.2.10.31/26
      Moveable: Immediate SrvMgr: No  Flg: 1
    IntfName: INTFNAM1
      IpAddr: 2001:db8::522:f103
        Moveable: Immediate  SrvMgr: n/a Flg:
    IntfName: INTFNAM2
      IpAddr: 2001:db8::522:f203
        Moveable: Immediate  SrvMgr: n/a Flg:
    IntfName: INTFNAMB
      IpAddr: 2001:0db8::522:f222/64
        Moveable: Immediate SrvMgr: No  Flg: 1
    IntfName: INTFNAMC
      IpAddr: 2001:0db8::522:f333/64
        Moveable: Immediate SrvMgr: No  Flg: 2C
    IntfName: INTFNAMD
      IpAddr: 2001:0db8::22:f334/64
        Moveable: Immediate SrvMgr: No  Flg: C

  VIPA Range:
    IpAddr/PrefixLen: 201.2.10.192/26
      Moveable: NonDisr   SAFName: RANGE1
    IpAddr/PrefixLen: 201.2.20.192/26
      Moveable: Disrupt
    IntfName: INTFNAM3
      IpAddr/PrefixLen: 2001:db8::522:f303/24
        Moveable: NonDisr   SAFName: RANGE2
```

```
VIPA Distribute:
  Dest:        201.2.10.11..n/a
    DestXCF:   ALL
    DistMethod: Roundrobin
    SysPt:   Yes  TimAff: 200 Flg: Roundrobin
    OptLoc:  No
  Dest:        201.2.10.13..243
    DestXCF:   ALL
    DistMethod: BaseWLM
    SysPt:   No   TimAff: No  Flg: OptLocal
    OptLoc:  1
  Dest:        201.2.10.14..244
    DestXCF:   ALL
    DistMethod: ServerWLM
    SysPt:   No   TimAff: No  Flg: Tier1
    OptLoc:  No
  Dest:        201.2.10.17..8080
    DestXCF:   200.1.10.10
    DistMethod: TargCtrl
    SysPt:   No   TimAff: Yes  Flg: Tier1
    OptLoc:  No
    GrpName: CICSGROUP       RtgType: GRE Control Port: 1010
  Dest:        201.2.10.18..4040
    DestXCF:   201.3.10.15
    DistMethod: ServerWLM
    SysPt:   No   TimAff: Yes  Flg: Tier2
    OptLoc:  No
    ProcXCost:
      zAAP: 001  zIIP: 001
    ILWeighting: 0
    GrpName: CICSGROUP
  Dest:        201.2.10.18..4040
    DestXCF:   201.3.10.16
    DistMethod: ServerWLM
    SysPt:   No   TimAff: Yes  Flg: Tier2
    OptLoc:  No
    ProcXCost:
      zAAP: 001  zIIP: 001
    ILWeighting: 0
    GrpName: CICSGROUP
  Dest:        201.4.10.15..7000
    DestXCF:   201.3.10.16
    DistMethod: HotStandby      SrvType: Preferred
    AutoSwitchBack: Yes         HealthSwitch: Yes
    SysPt:   No   TimAff: Yes   Flg:
    OptLoc:  No
  Dest:        201.4.10.15..7000
    DestXCF:   201.3.10.17
    DistMethod: HotStandby      SrvType: Backup  Rank: 001
    AutoSwitchBack: Yes         HealthSwitch: Yes
    SysPt:   No   TimAff: Yes   Flg:
    OptLoc:  No
  DestIntf:    INTFNAM1
    Dest:      2001:db8::522:f103..20
      DestXCF: ALL
      DistMethod: ServerWLM
      SysPt:   No   TimAff: No  Flg:
      OptLoc:  No
      ProcXCost:
        zAAP: 001  zIIP: 001
      ILWeighting: 0
  DestIntf:    INTFNAM1
    Dest:      2001:db8::522:f103..21
      DestXCF: ALL
      DistMethod: ServerWLM
      SysPt:   Yes  TimAff: 10  Flg:
      OptLoc:  No
      ProcXCost:
        zAAP: 001  zIIP: 001
      ILWeighting: 0
```

```
DestIntf:    INTFNAMB
      Dest:      2001:0db8::522:f222..8000
        DestXCF: 2001:0db8::540:f301
        DistMethod: TargCtrl
        SysPt:   No   TimAff: No  Flg: Tier1
        OptLoc:  No
        GrpName: FTPGROUP2        RtgType: ENCAP CtrlPort: 1010
    DestIntf:    INTFNAMB
      Dest:      2001:0db8::522:f222..8000
        DestXCF: 2001:0db8::540:f302
        DistMethod: TargCtrl
        SysPt:   No   TimAff: No  Flg: Tier1
        OptLoc:  No
        GrpName: FTPGROUP2        RtgType: ENCAP CtrlPort: 1010

  VIPA Service Manager:
    McastGroup: 224.0.0.1
    Port: 04444  Pwd: Yes

  VIPA Route:
    DestXCF:     201.10.10.1
      TargetIp:  201.20.20.1
    DestXCF:     201.10.10.2
      TargetIp:  201.20.20.2
    DestXCF:     2eco::500:f103
      TargetIp:  2eco::100:f103

Deactivated Dynamic VIPA Information:
VIPA Backup:
    IpAddr/PrefixLen: 201.2.10.40/26
      Rank: 025  Moveable: Immediate  SrvMgr: No  Flg:

  VIPA Define:
    IpAddr/PrefixLen: 201.2.10.20/26
      Moveable: Immediate  SrvMgr: No  Flg:

  VIPA Distribute:
    Dest:        201.2.10.20..5000
      DestXCF:   ALL
        SysPt:   No   TimAff: No   Flg: BaseWLM
```

Report field descriptions:
Displays the following dynamic VIPA information defined in the VIPADYNAMIC profile statement. For more information about each field, see the VIPADYNAMIC profile statements in the z/OS Communications Server: IP Configuration Reference.

VIPA Backup

Displays the following configured dynamic VIPA backup information:

For a SHORT format report:

IP Address

The IP address for this DVIPA.

AddressMask

The net mask that determines how many of the bits of the IP address determine the net. This field is blank if Moveable and AddressMask were not specified on the VIPABACKUP statement or if another stack initially activated the DVIPA.

For a LONG format report:

IntfName

The name of this IPv6 interface. This name will match the interface name defined on the Primary stack that is being backed up.

IpAddr/PrefixLen
The IP address and prefix length for this DVIPA. For an IPv4 address, the prefix length range is 1 - 32. For an IPv6 address, the prefix length range is 1 – 128.

For a SHORT or LONG format report:

Rank The relative position of this stack in the list of stacks that can activate (takeover) the DVIPA in case of failure. The stack with the highest ranked backup DVIPA will do the takeover.

Moveable
Indicates the conditions under which the active DVIPA can be moved to another stack. This field is blank if Moveable and AddressMask were not specified on the VIPABACKUP statement, or if another stack initially activated the DVIPA.

WhenIdle
Indicates that this DVIPA can be moved to another stack when there are no connections for this DVIPA on the current stack. If there are connections on the current stack at the time another stack issues a VIPADEFINE for the same DVIPA, the DVIPA remains active on this stack until the last connection on this stack ends.

Immediate
Indicates that this DVIPA can be moved to another stack as soon as the other stack requests ownership by executing a VIPADEFINE for the same DVIPA. Existing connections on the current stack are maintained by the new owning stack.

SrvMgr
Indicates whether sysplex distributor performs Multinode Load Balancing (MNLB) by functioning as a Service Manager (in place of Cisco's LocalDirector) for this DVIPA. This field for an IPv4 entry is blank if Moveable and AddressMask were not specified on the VIPABACKUP statement or if another stack initially activated the DVIPA. This field for an IPv6 entry will always display **n/a** as it is not applicable for IPv6.

Flg The following values can be displayed in the Flg field:

1 Indicates that this DVIPA is used to distribute incoming requests to z/OS or non-z/OS targets (for example, DataPower appliances).

2 Indicates that this DVIPA is used to distribute incoming requests from tier 1 targets to the group of server applications.

C Indicates that this DVIPA is specific to the central processor complex (CPC) that it is defined on and that it cannot be moved to or taken over by a TCP/IP stack on a different CPC. This DVIPA can serve as a default route from DataPower appliances that are associated with this CPC. When used with the tier 2 flag (see Flg value 2), this value indicates that all of the tier 2 target applications are on TCP/IP stacks on the same CPC.

VIPA Define
Displays the configured dynamic VIPA define information.

For a SHORT format report:

IP Address

The IP address for this DVIPA.

AddressMask

The net mask that determines how many of the bits of the IP address determine the net.

For a LONG format report:

IntfName

The name of this IPv6 interface.

IpAddr/PrefixLen

The IP address and prefix length for this DVIPA. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

For a SHORT or LONG format report:

Moveable

Indicates the conditions under which the DVIPA can be moved to another stack.

WhenIdle

Indicates that this DVIPA can be moved to another stack when there are no connections for this DVIPA on the current stack. If there are connections on the current stack at the time another stack issues a VIPADEFINE for the same DVIPA, the DVIPA remains active on this stack until the last connection on this stack ends.

Immediate

Indicates that this DVIPA can be moved to another stack as soon as the other stack requests ownership by executing a VIPADEFINE for the same DVIPA. Existing connections on the current stack are maintained by the new owning stack.

SrvMgr

Indicates whether sysplex distributor performs Multinode Load Balancing (MNLB) by functioning as a Service Manager (in place of Cisco's LocalDirector) for this DVIPA. This field for an IPv6 entry always displays **n/a** as it is not applicable for IPv6.

Flg

The following values can be displayed in the Flg field:

1

Indicates that this DVIPA is used to distribute incoming requests to the z/OS targets or non-z/OS targets (for example, DataPower appliances).

2

Indicates that this DVIPA is used to distribute incoming requests from tier 1 targets to the group of server applications.

C

Indicates that this DVIPA is specific to the central processor complex (CPC) that it is defined on and that it cannot be moved to or taken over by a TCP/IP stack on a different CPC. This DVIPA can serve as a default route from DataPower appliances that are associated with this CPC. When used with the tier 2 flag (see Flg value 2), this value indicates that all of the tier 2 target applications are on TCP/IP stacks on the same CPC.

VIPA Range

Displays the configured dynamic VIPA range information.

For a SHORT format report:

AddressMask

The net mask that determines how many bits of the IP address determine the net.

IP Address

An IP address that determines a VIPARANGE net value when ANDed with the specified address mask. DVIPAs that fall within the range can be created by BIND or SIOCSVIPA ioctl.

SAFName

The final qualifier of a System Authorization Facility (SAF) resource name. The maximum length is 8 characters.

For a LONG format report:

IntfName

The name of this IPv6 interface.

IpAddr/PrefixLen

The IP address and prefix length for this DVIPA. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

SAFName

The final qualifier of a System Authorization Facility (SAF) resource name. The maximum length is 8 characters.

For a SHORT or LONG format report:

Moveable

Indicates the conditions under which DVIPAs created within this VIPARANGE can be moved to another stack.

Disrupt

Indicates that nondisruptive movement will not occur for DVIPAs created within this VIPARANGE on this stack. In the case of a BIND-created DVIPA, a subsequent BIND for the same DVIPA will not move the DVIPA and the subsequent BIND will fail. In the case of an ioctl-created DVIPA, a subsequent ioctl request for the same DVIPA moves the DVIPA to the new stack, but connections on that DVIPA on the first stack are broken.

NonDisr

Indicates immediate nondisruptive movement for DVIPAs within this VIPARANGE created on this stack by SIOCSVIPA ioctl (or BIND) when the same DVIPA is requested by subsequent net mask (or subsequent BIND) on another stack. Any existing connections on the original owning stack are maintained by the new owning stack.

VIPA Distribute

Displays the configured dynamic VIPA define information.

For a SHORT format report:

IP Address
The specific IP address for which incoming connections are to be distributed.

Port
The specific port for which incoming connections are to be distributed. A port value of n/a indicates that the PORT parameter was not specified on the VIPADISTRIBUTE profile statement.

Result: If multiple ports were specified individually or in a range on a VIPADISTRIBUTE statement, one entry is displayed for each address and port combination.

XCF Address
The dynamic XCF address (IPCONFIG DYNAMICXCF) of a target stack for incoming connections to the DVIPA and port.

If the Flg field value 1 is displayed and RtgType `GRE` is displayed when the DETAIL keyword is used, then this field represents the IP address of the tier 1 target for incoming connections to the DVIPA.

Weight
The configured distribution method is WEIGHTEDActive. This is the configured active connection weight that is used when incoming connections are distributed to this target stack.

Flg
Flags including the following values:

1
Indicates that the DVIPA is used to distribute incoming requests to tier 1 targets.

2
Indicates that the DVIPA is used to distribute incoming requests from tier 1 targets to a group of tier 2 server applications.

O
Indicates that the OPTLOCAL keyword was defined on the VIPADISTRIBUTE profile statement. To see the OPTLOCAL value currently in effect, issue the Netstat VIPADCFG/**-F** command with the DETAIL keyword.

For a LONG format report:

DestIntf
The name of this IPv6 interface.

Dest
The specific IP address and port for which incoming connections are to be distributed. A port value of n/a indicates that the PORT parameter was not specified on the VIPADISTRIBUTE profile statement.

Result: If multiple ports were specified individually or in a range on a VIPADISTRIBUTE statement, one entry is displayed for each address and port combination.

DestXCF
The dynamic XCF address (IPCONFIG6 DYNAMICXCF) of a target stack for incoming connections to the DVIPA and port.

If the Flg field value 1 is displayed and the RtgType field value `GRE` or `ENCAP` is displayed when the DETAIL keyword is used, then this field represents the IP address of the tier 1 target for incoming connections to the DVIPA.

Flg
Flags including the following values:

OptLocal
Indicates that the OPTLOCAL keyword was defined on the VIPADISTRIBUTE profile statement. To see the OPTLOCAL value currently in effect, issue the Netstat VIPADCFG/**-F** command with the DETAIL keyword.

Tier1 Indicates that the DVIPA is used to distribute incoming requests to tier 1 targets.

Tier2 Indicates that the DVIPA is used to distribute incoming requests from tier 1 targets to a group of tier 2 server applications.

For a SHORT or LONG format report:

DistMethod
The distribution method in use.

BaseWLM
Indicates that WLM system weights and policy information are used to distribute incoming connection requests.

HotStandby
Indicates that HotStandby distribution is in use.

Roundrobin
Indicates that incoming connection requests are distributed using the round-robin method.

ServerWLM
Indicates that WLM server weights and policy information are used to distribute incoming connection requests.

WLM server weights are used if SERVERWLM was specified on the VIPADISTRIBUTE statement for this DVIPA and port, and all target servers are able to provide WLM server-specific weights. Otherwise, BaseWLM is used.

WeightedActive
Indicates that incoming connection requests are distributed using the weighted active connections method.

Rule: For a short format report, you need to use the DETAIL modifier to display this field.

SrvType
Indicates the server type when the HotStandby distribution method is configured.

Backup
Indicates that this is a backup target. A backup target is initially a hot standby target. Connections are not distributed to hot standby targets. If the active target becomes unavailable, the distributor switches targets and one of the hot standby targets becomes the active target.

When the server type is backup, the rank of the backup is displayed:

Rank Used to determine which target is selected if the preferred target is unavailable. The highest ranked available backup is used.

Preferred

Indicates that this is the preferred target. If AUTOSWITCHBACK is configured, then the preferred target is the active target if it is available and has not had any health problems. If the active target becomes unavailable, the distributor switches to use a hot standby target; the active target becomes a hot standby target and the selected hot standby target becomes the active target.

Rule: For a short format report, you need to use the DETAIL modifier to display this field.

SysPt Indicates whether coordinated Sysplex-wide ephemeral port assignment is activated for this distributed DVIPA.

TimAff

The value that was defined in the TIMEDAFFINITY parameter on the VIPADISTRIBUTE profile statement. The value No indicates that TIMEDAFFINITY is not specified or is set to zero.

DETAIL

Displays the general dynamic VIPA configuration information and the OPTLOCAL value.

If the distribution method is WEIGHTEDActive, then the configured active connection weight is displayed.

If the distribution method is HotStandby, then the AUTOSWITCHBACK and HEALTHSWITCH information is displayed.

If the DVIPA value TIER1 or TIER2 was specified, the name of the targeted server application group is displayed.

AUTOSWITCHBACK

Indicates whether the distributor automatically switches distribution back to the preferred target when it is available and healthy.

If the preferred target becomes a standby target because it is no longer available and later becomes available and healthy, the value `Yes` indicates that the distributor automatically switches back to using the preferred target as the active target. This is the default value.

The value `No` indicates that the distributor does not automatically switch back to the preferred target.

HEALTHSWITCH

Indicates whether the distributor automatically switches from the active target if the active target is not healthy.

The value `Yes` indicates that the distributor does switch from the active target when it is not healthy. This is the default.

The value `No` indicates that the distributor ignores health metrics. The distributor switches from the active target only if the target is not ready or if the distributor does not have an active route to the target.

OPTLOCAL

A value of 0 indicates that connections originating from a target stack within the sysplex should always bypass sending the connection request to the sysplex distributor. The relative capacity of the WLM weights for servers on other target stacks within the sysplex are not considered when determining whether the connection should remain local.

A value of 1 indicates that connections originating from a target stack within the sysplex should always bypass sending the connection request to the sysplex distributor as long as the WLM weight for the server on the local target stack's WLM weight is not 0. This is the default value if the OPTLOCAL field is specified without a value.

If a value in the range 2 – 16 is specified, this value is used as a multiplier against the raw WLM weight of the server on the local target stack to cause this server to be favored over the servers on other target stacks. The relative capacity of the WLM weights of the servers on the other target stacks within the sysplex is considered when determining which stack should process the connection. The greater the value specified, the more likely that the local stack is favored over other target stacks.

Regardless of the value that is specified on the OPTLOCAL statement, if one of the following conditions exists, connections are sent to the distributing stack:

- No local server is available
- The SEF value has fallen below 75
- The number of abnormal transaction completions has exceeded 250
- The health indicator is less than 75

Weight

The configured distribution method is WEIGHTEDActive. This is the configured active connection weight that is used when incoming connections are distributed to this target stack.

PROCTYPE

The expected utilization proportion of each type of processor (CP, zAAP, and zIIP) that an application's workload will consume. This field is displayed only when the configured distribution method is BASEWLM.

CP The expected utilization proportion of general CPU processor capacity.

zAAP The expected utilization proportion of zAAP processor capacity.

zIIP The expected utilization proportion of zIIP processor capacity.

ProcXCost

The crossover cost that is applied to the workload that was targeted to a zAAP or zIIP processor but that ran on the conventional processor when the composite SERVERWLM weight is determined. The weight 1 indicates that crossover cost is not considered when the composite SERVERWLM weight is determined. This parameter is displayed only when the distribution method is SERVERWLM.

zAAP The crossover cost of running a workload that was targeted to a zAAP processor on a general CPU instead of on the zAAP processor.

zIIP The crossover cost of running a workload that was targeted to a zIIP processor on a general CPU instead of on the zIIP processor.

ILWeighting
The configured importance level weighting factor. This field is displayed only when the configured distribution method is SERVERWLM. See the following possible values:

0 Indicates that systems that have displaceable capacity at low importance levels are not favored over systems that have displaceable capacity at high importance levels.

1 Indicates that WLM should weight displaceable capacity at each successively lower importance level slightly higher than the capacity that is at the preceding higher importance level. The weighting increases proportionally to the square root of the difference between the importance level values plus 1. This ILWeighting value provides a moderate bias when you compare displaceable capacity at different importance levels.

2 Indicates that WLM should weight displaceable capacity at each successively lower importance level significantly higher than the capacity that is at the preceding higher importance level. The weighting increases proportionally to the difference between the importance level values plus 1. This ILWeighting value provides an aggressive bias when you compare displaceable capacity at different importance levels.

3 Indicates that WLM should weight displaceable capacity at each successively lower importance level significantly higher than the capacity that is at the preceding higher importance level. The weighting grows proportionally to the square of the difference between the importance level values plus 1. This ILWeighting value provides an exceptionally aggressive bias when you compare displaceable capacity at different importance levels.

GrpName
The name of the targeted server application group, if TIER1 or TIER2 was specified for this DVIPA. This name is used to correlate the two tiers of sysplex distribution to and from tier 1 targets.

RtgType
Indicates the protocol that is used when requests are routed to the tier 1 targets.

GRE Indicates that IPv4 Generic Routing Encapsulation (GRE) is used to forward requests.

ENCAP
Indicates that IPv6 encapsulation is used to forward requests.

CtrlPort
Specifies the port number to be used for the control connection with the tier 1 target.

VIPA Service Manager
Displays the configured dynamic VIPA service manager information.

McastGroup
The multicast address used for communications between the sysplex distributor and the Cisco routers acting as forwarding agents.

Port The UDP port used for communications between the sysplex distributor and Cisco forwarding agents.

PWD Indicates whether the SMPASSWORD was specified.

VIPA Route
Displays the configured route information defined by the VIPAROUTE statement.

For a SHORT format report:

XCF Address
The dynamic XCF address (IPCONFIG DYNAMICXCF) of a target stack.

TargetIp
The IP address in the HOME list of the target stack that should be used to obtain the best available route from the sysplex distributor to that target.

For a LONG format report:

DestXCF
The dynamic XCF address (IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF) of a target stack.

TargetIp
The IP address in the HOME list of the target stack that should be used to obtain the best available route from the sysplex distributor to that target.

Deactivated Dynamic VIPA Information
Displays the configured VIPABACKUP, VIPADEFINE, and VIPADISTRIBUTE definitions that have been deactivated by the VARY TCPIP,,SYSPLEX,DEACTIVATE,DVIPA= command. See “VARY TCPIP,,SYSPLEX” on page 265 for more information about the command.

Netstat VIPADyn/-v report

Displays the current dynamic VIPA and VIPAROUTE information for a local host.

TSO syntax:

```
►►──NETSTAT VIPADyn──┤ Modifier ├──┤ Target ├──┤ Output ├──────────────────►◄
```

Modifier:

```
►►──┬─DVIPA─────┬──────────────────────────────────────────────────────────►◄
    └─VIPAROUTE─┘
```

DVIPA
: Displays the current dynamic VIPA information only.

VIPAROUTE
: Displays the current VIPAROUTE information only.

Target:
Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 306 or "Netstat command output" on page 316.

z/OS UNIX syntax:

```
►►──netstat -v──┤ Modifier ├──┤ Target ├──┤ Output ├───────────────────────►◄
```

Modifier:

```
►►──┬─DVIPA─────┬──────────────────────────────────────────────────────────►◄
    └─VIPAROUTE─┘
```

DVIPA
: Displays the current dynamic VIPA information only.

VIPAROUTE
: Displays the current VIPAROUTE information only.

Target:
Provide the report for a specific TCP/IP address space by using **-p** *tcpname*. See "The Netstat command target" on page 316 for more information about the TCp parameter.

Output:
The default output option displays the output to z/OS UNIX shell stdout. For other options, see "The z/OS UNIX netstat command syntax" on page 310 or "Netstat command output" on page 316.

Command syntax examples:

From TSO environment:

```
NETSTAT VIPADYN
   Display the current dynamic VIPA and VIPAROUTE information for a local host in the default
   TCP/IP stack.
NETSTAT VIPADYN DVIPA
   Display the current dynamic VIPA information for a local host in the default TCP/IP stack.
NETSTAT VIPADYN VIPAROUTE
   Display the current VIPAROUTE information for a local host in the default TCP/IP stack.
NETSTAT VIPADYN TCP TCPCS6
   Display the current dynamic VIPA and VIPAROUTE information for a local host in the TCPCS6
   stack.
```

From UNIX shell environment:

```
   netstat -v
   netstat -v DVIPA
   netstat -v VIPAROUTE
   netstat -v -p tcpcs6
```

Report examples:
The following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX **netstat** command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format):

```
NETSTAT VIPADYN

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           18:28:50
Dynamic VIPA:
  IP Address      AddressMask     Status    Origination     DistStat
  ----------      -----------     ------    -----------     --------
  201.2.10.11     255.255.255.192 Active    VIPADefine      Dist
    ActTime:      03/02/2005 16:45:20
  201.2.10.12     255.255.255.192 Active    VIPADefine      Dist/Dest
    ActTime:      03/02/2005 16:45:20
  201.2.10.14     255.255.255.192 Backup    VIPABackup
    ActTime:      n/a
  201.2.10.32     <None>          Backup    VIPABackup
    ActTime:      n/a
  199.199.199.8   255.255.255.0   ACTIVE    VIPARANGE IOCTL
    ActTime:      03/02/2005 16:45:20       JobName:        JOBTST1A
    Affinity:     No
  199.199.199.9   255.255.255.0   ACTIVE    VIPARANGE BIND
    ActTime:      03/02/2005 16:45:20       JobName:        JOBTST1B


VIPA Route:
  XCF Address     TargetIp         RtStatus
  -----------     --------         --------
  201.10.10.1     201.20.20.1      Defined
  201.10.10.2     201.20.20.2      Active
  201.10.10.3     201.20.20.3      Unavail


NETSTAT VIPADYN DVIPA

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           18:28:50
Dynamic VIPA:
  IP Address      AddressMask     Status    Origination     DistStat
  ----------      -----------     ------    -----------     --------
  201.2.10.11     255.255.255.192 Active    VIPADefine      Dist
    ActTime:      03/02/2005 16:45:20
  201.2.10.12     255.255.255.192 Active    VIPADefine      Dist/Dest
    ActTime:      03/02/2005 16:45:20
  201.2.10.14     255.255.255.192 Backup    VIPABackup
    ActTime:      n/a
  201.2.10.32     <None>          Backup    VIPABackup
    ActTime:      n/a
  199.199.199.8   255.255.255.0   ACTIVE    VIPARANGE IOCTL
    ActTime:      03/02/2005 16:45:20       JobName:        JOBTST1A
    Affinity:     No
  199.199.199.9   255.255.255.0   ACTIVE    VIPARANGE BIND
    ActTime:      03/02/2005 16:45:20       JobName:        JOBTST1B


NETSTAT VIPADYN VIPAROUTE

MVS TCP/IP NETSTAT CS V2R1       TCPIP Name: TCPCS           18:28:50
VIPA Route:
  XCF Address     TargetIp         RtStatus
  -----------     --------         --------
  201.10.10.1     201.20.20.1      Defined
  201.10.10.2     201.20.20.2      Active
  201.10.10.3     201.20.20.3      Unavail
```

IPv6 enabled or request for LONG format:

```
NETSTAT VIPADYN
MVS TCP/IP NETSTAT CS V2R1      TCPIP Name: TCPCS           18:29:44
Dynamic VIPA:
   IpAddr/PrefixLen: 201.2.10.11/26
     Status:  Active     Origin: VIPADefine       DistStat: Dist
     ActTime: 03/02/2005 16:45:20
   IpAddr/PrefixLen: 201.2.10.12/26
     Status:  Active     Origin: VIPADefine       DistStat: Dist/Dest
     ActTime: 03/02/2005 16:45:20
   IpAddr/PrefixLen: 201.2.10.14/26
     Status:  Backup     Origin: VIPABackup       DistStat:
     ActTime: n/a
   IpAddr/PrefixLen: 201.2.10.32
     Status:  Backup     Origin: VIPABackup       DistStat:
     ActTime: n/a
   IPADDR/PREFIXLEN: 199.199.199.8/24
     Status:  Active     Origin: VIPARange IOCTL  Affinity: No
     ActTime: 03/02/2005 16:45:20                 JobName:  JOBTST1A
   IPADDR/PREFIXLEN: 199.199.199.9/24
     Status:  Active     Origin: VIPARange BIND
     ActTime: 03/02/2005 16:45:20                 JobName:  JOBTST1B
   IntfName: INTFNAM1
     IpAddr: 2001:0db8::522:f103
       Status:  Active     Origin: VIPADefine       DistStat: Dist/Dest
       ActTime: 03/02/2005 16:45:20
   IntfName: INTFNAM2
     IpAddr: 2001:0db8::522:f203
       Status:  Active     Origin: VIPADefine       DistStat:
       ActTime: 03/02/2005 16:45:20
   IntfName: INTFNAMR1
     IpAddr: 2001:0db8::522:f229
       Status:  Active     Origin: VIPARange IOCTL  Affinity: No
       ActTime: 03/02/2005 16:45:20                 JobName:   JOBTST6A


VIPA Route:
  DestXCF:     201.10.10.1
    TargetIp:  201.20.20.1
    RtStatus:  Defined
  DestXCF:     201.10.10.2
    TargetIp:  201.20.20.2
    RtStatus:  Active
  DestXCF:     2eco::500:f103
    TargetIp:  2eco::100:f103
    RtStatus:  Unavail
```

```
NETSTAT VIPADYN DVIPA

MVS TCP/IP NETSTAT CS V2R1        TCPIP Name: TCPCS           18:29:44
Dynamic VIPA:
  IpAddr/PrefixLen: 201.2.10.11/26
    Status:  Active     Origin: VIPADefine       DistStat: Dist
    ActTime: 03/02/2005 16:45:20
  IpAddr/PrefixLen: 201.2.10.12/26
    Status:  Active     Origin: VIPADefine       DistStat: Dist/Dest
    ActTime: 03/02/2005 16:45:20
  IpAddr/PrefixLen: 201.2.10.14/26
    Status:  Backup     Origin: VIPABackup       DistStat:
    ActTime: n/a
  IpAddr/PrefixLen: 201.2.10.32
    Status:  Backup     Origin: VIPABackup       DistStat:
    ActTime: n/a
  IPADDR/PREFIXLEN: 199.199.199.8/24
    Status:  Active     Origin: VIPARange IOCTL  Affinity: No
    ActTime: 03/02/2005 16:45:20                 JobName:  JOBTST1A
  IPADDR/PREFIXLEN: 199.199.199.9/24
    Status:  Active     Origin: VIPARange BIND
    ActTime: 03/02/2005 16:45:20                 JobName:  JOBTST1B
  IntfName: INTFNAM1
    IpAddr: 2001:0db8::522:f103
      Status:  Active     Origin: VIPADefine       DistStat: Dist/Dest
      ActTime: 03/02/2005 16:45:20
  IntfName: INTFNAM2
    IpAddr: 2001:0db8::522:f203
      Status:  Active     Origin: VIPADefine       DistStat:
      ActTime: 03/02/2005 16:45:20
  IntfName: INTFNAMR1
    IpAddr: 2001:0db8::522:f229
      Status:  Active     Origin: VIPARange IOCTL  Affinity: No
      ActTime: 03/02/2005 16:45:20                 JobName:   JOBTST6A

NETSTAT VIPADYN VIPAROUTE

MVS TCP/IP NETSTAT CS V2R1        TCPIP Name: TCPCS           18:29:44
 VIPA Route:
  DestXCF:     201.10.10.1
    TargetIp:  201.20.20.1
    RtStatus:  Defined
  DestXCF:     201.10.10.2
    TargetIp:  201.20.20.2
    RtStatus:  Active
  DestXCF:     2eco::500:f103
    TargetIp:  2eco::100:f103
    RtStatus:  Unavail
```

Report field descriptions:

For a SHORT format report:

IP Address
The IP address for this DVIPA.

AddressMask
The net mask that determines how many of the bits of the IP address determine the net.

For a LONG format report:

IntfName
The name of this IPv6 interface.

IpAddr/PrefixLen
The IP address and prefix length for this DVIPA. For an IPv4 address, the prefix length range is 1 – 32. For an IPv6 address, the prefix length range is 1 – 128.

For a SHORT or LONG format report:

Dynamic VIPA
Displays the current dynamic VIPA information.

Status The state of the DVIPA on this stack. It can be any one of the following value:

Active The DVIPA is active on this stack.

Backup
This stack is eligible to activate the DVIPA if the stack where the DVIPA is currently active goes down or deletes the DVIPA.

Tip: If the DistStat value is equal to Dest, then the DVIPA is currently a target for distribution.

Moving
The DVIPA was active on this stack and has been moved to another stack. The DVIPA remains in 'Moving' status no established connections exist. Connections on this stack for this DVIPA that were established before the move are being serviced. If new connections on this stack use the DVIPA, they are also serviced.

Quiescing
The DVIPA was a target for distribution and has been removed as a target. However, connections for this DVIPA are still being serviced. The DVIPA is removed from this stack when all its connections complete.

Origin
Indicates how the DVIPA was created. It can be one of the following value:

VIPABackup
The DVIPA was created with a VIPABACKUP profile statement.

VIPADefine
The DVIPA was created with a VIPADEFINE profile statement.

VIPARange Bind
The DVIPA was created when a socket did an explicit bind to an IP address that fell with a range of IP addresses configured on a VIPARANGE profile statement.

VIPARange ioctl
The DVIPA was created when an application, or the MODDVIPA utility, issued an SIOCSVIPA or SIOCSVIPA6 ioctl to create a DVIPA that was within a range of IP addresses configured on a VIPARANGE profile statement

Blank The DVIPA was not explicitly created on this stack. It was dynamically created when another stack processed a

VIPADISTRIBUTE statement that specified this stack to be a target for connections to this DVIPA.

DistStat

Indicates that the distribution status for this DVIPA. It can be one of the following value:

Dist This stack is distributing incoming connections for the DVIPA to one or more other stacks in the sysplex.

Dist/Dest
This stack is distributing incoming connections for this DVIPA to one or more stacks in the sysplex and this stack is also a target for the distribution.

Dest The DVIPA was activated on this stack because this stack is a target for distributed connections to this DVIPA.

Blank The DVIPA is neither being distributed by this stack, nor a target of distribution from another stack.

Rule: `DistStat` is not used if the Origin is VIPARANGE.

Affinity

Indicates whether a DVIPA with an origin of VIPARANGE IOCTL was created with affinity. A connection request for a DVIPA that was created with affinity is sent to a TCP listener if its bind() call was issued by the application instance that created the DVIPA. If no matching listener is found, a TCP listener is selected by using normal shareport load balancing.

No A DVIPA was not created with affinity.

Yes A DVIPA was created with affinity.

ActTime

The time when this DVIPA was activated on the local stack, either because it is the owner of the DVIPA or because it is a target for this DVIPA, specified as Coordinated Universal Time (UTC).

The value `n/a` indicates that this DVIPA was not owned by this stack or that this stack is not the target for distributed connections to this DVIPA.

JobName

The job name of either the application or the MODDVIPA utility that enabled creation of this DVIPA. This field is significant only when this DVIPA was created with one of the following methods:

- A socket performed an explicit bind to an IP address that fell within a range of IP addresses configured on a VIPARANGE profile statement.
- An application or the MODDVIPA utility issued an SIOCSVIPA or SIOCSVIPA6 ioctl call to create a DVIPA that fell within a range of IP addresses configured on a VIPARANGE profile statement.

The environment in which the application runs determines the job name that is to be associated with a particular client or server application. The following list explains how to determine the JobName value, given the environment in which the application is run:

- Applications submitted as batch jobs use the batch job name.
- The job name associated with applications that are started from the MVS operator console using the START command is determined as follows:
  - If the START command is issued with the name of a member in a cataloged procedure library (for example, S APP1), then the job name is the member name (for example, APP1).
  - If the member name on the START command is qualified by a started task identifier (for example, S APP1.ABC), then the job name is the started task identifier (for example, ABC).

  The JOBNAME parameter can also be used on the START command to identify the job name (for example, S APP1,JOBNAME=XYZ).

  The JOBNAME value can also be included on the JOB card.
- Applications that are run from a TSO user ID use the TSO user ID as the job name.
- Applications that run from the z/OS shell usually have a job name that is the logged on user ID plus a 1-character suffix.
- Authorized users can run applications from the z/OS shell and use the _BPX_JOBNAME environment variable to set the job name. In this case, the value specified for the environment variable is used as the job name.
- z/OS UNIX applications started by INETD typically use the job name of the INETD server plus a 1-character suffix.

VIPA Route

Displays the current VIPAROUTE information.

XCF Address or DestXCF

The dynamic XCF address (IPCONFIG DYNAMICXCF or IPCONFIG6 DYNAMICXCF) of a target stack.

TargetIp

The IP address in the HOME list of the target stack that should be used to obtain the best available route from the sysplex distributor to that target.

RtStatus

Indicates the status of the route entry. Can have the following values:

Active Indicates that the target stack identified by XCF Address or DestXCF is active, that *TargetIp* is defined at that target stack, and at least one route is available to *TargetIp*. The local stack will forward DVIPA packets to the target stack using normal IP routing table to determine the best available route.

Defined

Indicates that the target stack identified by XCF Address or DestXCF is not active or that the target stack is the same as the stack on which the VIPAROUTE is defined.

Inactive

Indicates that the target stack identified by XCF Address or DestXCF is active and that *TargetIp* is defined at that target stack; however no route is available to *TargetIp*. As a result, the local stack cannot forward any DVIPA packets to the target stack. For more information, see the steps for diagnosing sysplex routing problems in the z/OS Communications Server: IP Diagnosis Guide.

Unavail

Indicates that the target stack identified by XCF Address or DestXCF is active, but that *TargetIp* is not defined at that target stack. The local stack will forward DVIPA packets to the target stack using dynamic XCF interfaces. Message EZD1173I is issued when the routing stack detects this condition.

To correct the problem take the following actions:

1. Verify that the VIPAROUTE statement specifies the correct dynamic XCF address and target IP address for the required target stack.
2. Verify that the target IP address is correctly defined in the HOME list of the target stack.

z/OS UNIX and TSO Netstat option comparison

The following table shows the equivalent z/OS UNIX and TSO command formats.

Table 17. z/OS UNIX and TSO Netstat command options

TSO option	z/OS UNIX option	Description
Report Options		
ALL	**-A**	Displays detailed information about TCP connections and UDP sockets, including some recently closed ones.
ALLConn	**-a**	Displays information for all TCP connections and UDP sockets, including some recently closed ones.
ARp	**-R**	Displays ARP cache information.
BYTEinfo	**-b**	Displays the byte-count information for each active TCP connection and UDP socket.
CACHinfo	**-C**	Displays statistics for TCP listening sockets uszing the Fast Response Cache Accelerator (FRCA).
CLients	**-e**	Displays information about local users of TCP/IP services (jobnames).
CONFIG	**-f**	Displays the TCP/IP configuration information.
COnn	**-c**	Displays the information about each active TCP connection and UDP socket.
DEFADDRT	**-l**	Displays the policy table for IPv6 default address selection.
DEvlinks	**-d**	Displays information about interfaces that are defined to the TCP/IP stack.
Gate	**-g**	Displays information about the stack routing table for IPv4 destinations.
HElp	-?	Displays help information for Netstat parameters.
Home	**-h**	Displays information about each home IP address and its associated link or interface name.
IDS	**-k**	Displays information about Intrusion Detection Services. Displays Neighbor Discovery cache information (IPv6 only).
ND	**-n**	Displays Neighbor Discovery cache information (IPv6 only).

Table 17. z/OS UNIX and TSO Netstat command options (continued)

TSO option	z/OS UNIX option	Description
PORTList	**-o**	Displays the reserved port list.
RESCACHE	**-q**	Displays resolver cache information
ROUTe	**-r**	Displays information about the stack routing table for IPv4 destinations and IPv6 destinations if stack is IPv6 enabled.
SLAP	**-j**	Displays QoS Policy statistics.
SOCKets	**-s**	Displays the information about each client using a socket application programming interface.
SRCIP	**-J**	Displays the configured information for all job-specific, source IP address designations on the target TCP/IP.
STATS	**-S**	Displays TCP/IP statistics for IP, ICMP, TCP and UDP protocols.
TELnet	**-t**	Displays information for TN3270 Telnet server connections.
TTLS	**-x**	Displays Application Transparent Transport Layer Security (AT-TLS) information.
Up	**-u**	Displays the date and time that TCP/IP was started and specifies whether it is IPv6 enabled or disabled.
VCRT	**-V**	Displays the dynamic VIPA Connection Routing Table used for sysplex distributor and moveable dynamic VIPA support.
VDPT	**-O**	Displays the dynamic VIPA Distribution Port Table information.
VIPADCFG	**-F**	Displays the dynamic VIPA configuration for a TCP/IP stack.
VIPADyn	**-v**	Displays the current dynamic VIPA and VIPAROUTE information for a TCP/IP stack.
Target		
TCp	**-p**	Displays information for a specified TCP/IP address space.
Output		
FORMat	**-M**	Displays Netstat report in a given format.
REPort	n/a	Causes the output to be stored in the data set userid.NETSTAT.option.
STACk	n/a	Causes the output to be placed in the TSO data stack.
Filter		
APPLD	**-G**	Filter the output of ALL/**-A**, ALLConn/**-a**, and COnn/**-c** reports using the application data.
APPLname	**-L**	Filter the output of the TELnet/**-t** report using the specified VTAM application name.
CLIent	**-E**	Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, CLient/**-e**, COnn/**-c**, SOCKets/**-s**, and TELnet/**-t** reports using the specified client name.
CONNType	**-X**	Filter the output of the ALLConn/**-a** and COnn/**-c** reports using the specified connection type.
DNSAddr	**-Q**	Filter the output of the RESCache/**-q** report using the specified DNS IP address.
HOSTName	**-H**	Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, COnn/**-c**, RESCache/**-q**, SOCKets/**-s**, TELnet/**-t**, and VCRT/**-V** reports using the specified host name.
INTFName	**-K**	Filter the output of the DEvlinks/**-d** and HOme/**-h** reports using the specified interface name.

Table 17. z/OS UNIX and TSO Netstat command options (continued)

TSO option	z/OS UNIX option	Description
IPAddr	**-I**	Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, COnn/**-c**, Gate/**-g**, ND/**-n**, RESCache/**-q**, ROUTe/**-r**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, VDPT/**-O**, and VIPADCFG/**-F** reports using the specified IP address.
IPPort	**-B**	Filter the output of the ALL/**-A**, ALLConn/**-a**, COnn/**-c**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, and VDPT/**-O** reports using the specified IP address and port number.
LUName	**-L**	Filter the output of the TELnet/**-t** report using the specified LU name.
NOTN3270	**-T**	Filter the output of the ALL/**-A**, ALLConn/**-a**, BYTEinfo/**-b**, CLient/**-e**, COnn/**-c**, and SOCKets/**-s** reports excluding TN3270 server connections.
POLicyn	**-Y**	Filter the output of the SLAP/**-j** report using the specified policy rule name.
POrt	**-P**	Filter the output of the ALL/**-A**, ALLConn/**-a**, COnn/**-c**, PORTList/**-o**, SOCKets/**-s**, TELnet/**-t**, VCRT/**-V**, and VDPT/**-O** reports using the specified port number.
SMCID	**-U**	Filter the output of the ALL/**-A**, ALLConn/**-a**, COnn/**-c**, and DEvlinks/**-d** reports using the specified SMC-R link or link group identifier.
Command		
DRop	**-D**	Terminates the socket end-point that is identified by the specified connection number.

Ping

The TSO PING and z/OS UNIX **ping** commands determine the accessibility of a foreign node.

The TSO PING command: Send an echo request

The TSO PING command sends an echo request to a foreign node (remote host) to determine whether the node is accessible.

When a response to a Ping command is received, the elapsed time is displayed. The time does not include the time spent communicating between the user and the TCP/IP address space.

For information about the remote Ping function, which enables a user at one host to determine the response time between two remote hosts using SNMP, see Chapter 7, “Managing TCP/IP network resources with SNMP,” on page 953.

Format

```
►►──PING──┬─ host_name──┬──────────────────┬─┬──────────────────────────────►◄
          │             └─(──┤ Option ├────┘ │
          ├─ Help────────────────────────────┤
          └─ ?───────────────────────────────┘
```

Option:

```
   ┌──────────────────────────┐
   ▼                          │
├────┬─Addrtype──┬─ipv4─┬─────┬──────────────────────────────────────────────┤
     │           └─ipv6─┘     │
     │         ┌─1────┐       │
     ├─Count───┼──────┼───────┤
     │         └─echo─┘       │
     ├─Intf── interface───────┤
     │          ┌─256───┐     │
     ├─Length───┼───────┼─────┤
     │          └─bytes─┘     │
     ├─NOName─────────────────┤
     ├─PMTU───┬─yes────┬──────┤
     │        └─ignore─┘      │
     ├─Srcip── srcip──────────┤
     ├─TCP── tcpname──────────┤
     │           ┌─10──────┐  │
     ├─Timeout───┼─────────┼──┤
     │           └─seconds─┘  │
     └─Verbose────────────────┘
```

Note: The minimum abbreviation for each parameter is shown in uppercase letters.

Parameters

host_name

Specifies the host to which you want to send the echo request. This must be an IP address or a host name that can be resolved. IPv4-mapped IPv6 addresses are not supported.

If the *host_name* value is specified as a host name (not an IP address) the command invokes the resolver to obtain an IP address for the host name. The command uses the first IP address that is returned by the resolver. The ADDRTYPE option can be used to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If the ADDRTYPE option is not specified, the INTF and SRCIP options can also be used to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If ADDRTYPE, INTF, or SRCIP are not specified, then the command does not request a specific type of IP address from the resolver, so both IPv4 and IPv6 IP addresses can be returned by the resolver.

When using IPv6 link-local addresses, you can provide scope information with the IP address or host name. To specify scope information, add a percent character (%) after the *host_name* value, followed by the scope information (usually an interface name). The examples that follow include an example of using the command with scope information. For a more complete explanation about the use of scope information, see the support for scope information in the z/OS Communications Server: IPv6 Network and Application Design Guide.

Guidelines:

- When you are running multiple TCP/IP stacks on the same MVS image and the interface name that is used as the scope information has been defined to multiple TCP/IP stacks, you must specify the TCP parameter to ensure that the correct stack is used to send the command's packets.

- Providing scope information on the *host_name* option has the same effect as specifying the local interface using the INTF option, although the INTF option covers a wider range of situations (scope information applies only to IPv6 link-local addresses). If both methods of providing scope information are used on the same command, the values provided for scope information on the *host_name* option and for the INTF interface option must represent the same local interface, otherwise the command fails.

Addrtype ipv4 | ipv6

Specifies the IP address type that the Resolver returns when resolving the host name to an IP address. The values for this option are not case sensitive.

ipv6

Specifies that only IPv6 IP addresses are returned from the Resolver when resolving the host name to an IP address.

ipv4

Specifies that only IPv4 IP addresses are returned from the Resolver when resolving the host name to an IP address.

If the ADDRTYPE option is not specified, see the description of the *host_name* parameter for information on how the *host_name* value is resolved to an IP address.

Count *echo*

Sets the number of echo requests that are sent to the host. If you do not specify the Count parameter, the default value 1 is used, unless the Verbose parameter is specified. When the Verbose parameter is specified but the Count parameter is not specified, the default value is 3.

If *echo* is not specified, an error occurs. The *echo* value must be in the range 0 - 2^{31} minus 1, which is 2147483647. If *echo* is 0, the Ping command sends echo requests continually. To stop the Ping command, press **PA1**.

Restriction: If you specify the Verbose parameter, you cannot specify the value 0 for the Count parameter.

Intf *interface*

Specifies the local interface, *interface*, over which the packets are sent. The interface is either a name with a maximum of 16 bytes from a LINK or INTERFACE profile statement, or the IP address of a local interface. IPv4-mapped IPv6 addresses are not supported. Local VIPA or LOOPBACK interfaces are not valid.

If the destination host is specified as a host name and the ADDRTYPE option is not specified, the address type of the *interface* value is used to determine whether the host name is resolved to an IPv4 or IPv6 IP address.

When this parameter is specified, Ping establishes affinity to either the default TCP/IP stack or the stack specified on the TCP parameter. The specified interface must be defined to the stack to which Ping establishes affinity. You must also ensure that a route exists to the destination using the specified interface. This can be any kind of route, including a default route. This parameter is independent of the SRCIP parameter used as the source IP address in the outbound packets.

Note: As a diagnostics aid in analyzing response times and path availability using a particular route, this parameter routes packets over specified interfaces regardless of the multipath settings in the IPCONFIG MULTIPATH or IPCONFIG6 MULTIPATH profile statement by bypassing the outbound path selection algorithm for the packets.

Restriction:

- You cannot specify scope information for the *interface* value.
- To specify an OSM interface for the parameter, the user ID must have RACF authority to use the interface. For more information about OSM interface authorization, see OSM Access Control in z/OS Communications Server: IP Configuration Guide.

`Length` *`bytes`*

Sets the number of data bytes for the echo request. If a *bytes* value is not specified, an error occurs. If you do not specify the Length parameter, the default value 256 is used. The number of bytes must be in the range 8 - 65 487. A minimum of 8 data bytes is needed for a time stamp value, which the Ping command uses to correlate echo requests to echo replies.

For IPv4 destinations, the total length of the outbound echo request packet includes the length of an IPv4 IP header (20 bytes), the length of an ICMP header (8 bytes), and the data length specified by the Length parameter. Depending on your TCP/IP stack configuration, the TCP/IP stack might add additional IP header options to the IP header created by the Ping command before the echo request packet is sent.

For IPv6 destinations, the total length of the outbound echo request packet includes the length of an IPv6 IP header (40 bytes), the length of an ICMPv6 header (8 bytes), and the data length specified by the Length parameter. Depending on your TCP/IP stack configuration, the TCP/IP stack might add additional IPv6 extension headers to the packet that is created by the Ping command, before the echo request packet is sent.

`NOName`

Specifies that the Ping command should not resolve IP addresses to host names for ICMP/ICMPv6 messages received because of path MTU problems. This parameter is in effect only if the PTMU parameter was also specified; otherwise it is ignored. Specifying this parameter results in the Ping command displaying only the IP address of the host where fragmentation is needed. For example:

```
Ping #n needs fragmentation at: ipaddress
```

`PMTU yes | ignore`

This parameter can be used for diagnosing path maximum transmission unit (MTU) problems in the network. It prevents the outbound echo request packets from being fragmented and specifies what kind of path MTU discovery support should be used with the Ping command. For IPv4, path MTU discovery support is enabled by specifying the PATHMTUDISCOVERY parameter on the IPCONFIG profile statement. For IPv6, path MTU discovery support is enabled by default. The values for this option are not case sensitive.

If the echo request packets need to be fragmented at the local host or in the network, the Ping command displays the host name and IP address of the host where fragmentation is required.

yes Specifies that the outbound echo request packets are not fragmented at the local host or in the network and that the MTU value, determined by path MTU discovery for the destination, is used.

- If path MTU discovery is enabled and has already determined an MTU value for the destination, and the length of the Ping echo request packet is larger than this MTU size, then the local TCP/IP stack does not send out the packet. In this case, The Ping command displays one of the local stack's IP addresses as the host address where fragmentation is needed, and the next-hop MTU value displayed by the Ping command is the current path MTU value to the destination. For Ping commands to IPv4 destinations, the Ping command processing itself does not cause path MTU discovery support to be triggered for the destination. For IPv4, only TCP processing causes path MTU discovery support to be triggered.

- If path MTU discovery is not enabled or has not already determined a path MTU value for the destination, and the Ping echo request packet exceeds the configured route MTU selected for this packet, then the local TCP/IP stack does not send out the packet. In this case, the Ping command displays one of the local stack's IP addresses as the address of the host where fragmentation is needed, and the next-hop MTU value displayed by the Ping command is that of the route selected for the Ping packet.
- If the Ping request fails because the echo request packet requires fragmentation at some point in the network, the Ping command displays the IP address where fragmentation is required and displays the next-hop MTU value, if it was provided.

ignore Specifies that the outbound echo request packets are not fragmented at the local host or in the network, and that any MTU values determined by path MTU discovery for the destination, are ignored.

- If path MTU discovery determines an MTU value for the destination, and the length of the Ping echo request packet is larger than this MTU size, specifying the value **ignore** causes the TCP/IP stack to ignore the path MTU value and attempt to send out the packet. As long as the echo request packet length does not exceed the configured route MTU that is selected for this packet, you can use the **ignore** value to determine where in the network the original MTU problem occurred. In this case, the Ping command displays the IP address where fragmentation needs to occur and displays the path MTU value, if it was provided.
- If the Ping echo request packet exceeds the configured route MTU selected for this packet, then the local TCP/IP stack does not send out the packet. In this case, the Ping command displays one of the local stack's IP addresses as the address of the host where fragmentation is needed and the next-hop MTU value displayed by the Ping command is that of the route selected for the Ping packet.

If the Ping command receives an ICMP/ICMPv6 error message indicating that an echo request packet requires fragmentation, the Ping command displays the following output based on this message:

```
Ping #n needs fragmentation at: host_name (ipaddress)
```

If the host name resolution fails, the Ping command displays the following output:

```
Ping #n needs fragmentation at: ipaddress (ipaddress)
```

You can use the NOName parameter to request that the Ping command display only the host IP address, without resolving it to a host name.

If the host returned the next-hop MTU size in the ICMP/ICMPv6 message, then this MTU size is also displayed:

```
Next-hop MTU size is nnnnn
```

If the MTU size is not displayed, you can use the Length parameter to vary the size of the echo request packet, to determine the MTU of the network.

MULTIPATH PERPACKET considerations: When the MULTIPATH PERPACKET parameter is in effect and equal-cost routes are configured to the Ping destination host, the smallest MTU value of all the equal-cost routes is used as the largest packet size that can be sent, even if some of the equal-cost routes could support a larger packet size.

Srcip *srcip*

Specifies the source IP address, *srcip*. You must specify this as an IP address and not a host name. IPv4-mapped IPv6 addresses are not supported. On hosts with more than one IP address, you can set the source address to the IP address for another one of the stack's interfaces. This can be a VIPA address.

If the destination host is specified as a host name and the ADDRTYPE option is not specified, the address type of the *srcip* value is used to determine whether the host name is resolved to an IPv4 or IPv6 IP address.

Restriction: You cannot specify scope information for the source IP address.

TCP *tcpname*

Specifies the name of the TCP/IP stack that is to be used.

The *tcpname* is an 8-byte procedure name that is used to start the TCP/IP stack. When the `S member.identifier` method of starting TCP/IP is used, the value specified for identifier must be used as *tcpname*. When this option is not specified and z/OS UNIX is configured for CINET, the CINET Prerouter selects the TCP/IP stack to which the request is routed.

Timeout *seconds*

Sets the number of seconds that the Ping command waits for a response. If you do not specify the Timeout parameter, the default of 10 seconds is used. If a *seconds* value is not specified, an error occurs. The number of seconds must be in the range 1 - 100.

Verbose

Provides additional details about the received echo replies and a statistics summary.

If you do not specify the Verbose and Count parameters, then the default count of echo requests is 1. If you specify the Verbose parameter without the Count parameter, then the default value is 3. If you specify both the Verbose and Count parameters, then the number of echo requests is the value that is specified in the Count parameter.

Restriction: If you specify the value 0 for the Count parameter, you cannot specify the Verbose parameter.

See the examples that follow for the format of the Ping output when the Verbose parameter is specified. See the response descriptions that follow for the explanation of the fields that are used in the verbose information.

Help or question mark (?)

Provides help information about the Ping command. You cannot place the HELP parameter on the Ping command line with other parameters.

Usage

- To stop or interrupt the Ping command, press the **PA1** or **ATTN** key.

- You can place more than one parameter on the Ping command line; however, the HELP parameter is an exception and cannot be placed on the Ping command line with other parameters.
- To authorize the Ping command to use RAW sockets, add the command name, PING, to the AUTHCMD NAMES section of the member IKJTSOxx of SYS1.PARMLIB. TSO user IDs with UNIX System Services superuser authority are able to execute the command even without this SYS1.PARMLIB modification. If Ping is not authorized to use RAW sockets, Ping will fail with message `EZZ3115I Unable to open RAW socket: EDC5139I Operation not permitted.` For other authorization considerations, see MVS-related considerations information in the z/OS Communications Server: IP Configuration Guide.

Restrictions:

- Ping commands to a remote host might fail if there is a firewall between the two systems, even if the host is reachable using other commands.
- Ping commands to a remote host might be unable to detect path MTU information if there is an IPSec tunnel at any point between the two systems, even if the host is reachable using other commands. For more information about Ping PMTU interactions with IPSec tunnels, see "Resolving TSO PING and z/OS UNIX ping command problems" on page 673.

Examples

- IPv4

```
ping mvs098
CS V2R1: Pinging host MVS098 (9.67.113.11)
Ping #1 response took 0.002 seconds.
```

- IPv6

```
ping linuxipv62.tcp
CS V2R1: Pinging host LINUXIPV62.TCP.raleigh.ibm.com
at IPv6 address 2001:0db8::1:9:67:114:44
Ping #1 response took 0.002 seconds.
```

- IPv4 with the value `ignore` specified for the PMTU parameter and fragmentation needed out in the network. The hosts in this IPv4 network do not provide a next-hop MTU value when sending the ICMP error message. This example represents a network where there are multiple network paths to the destination.

```
ping hosta (count 4 pmtu ignore length 2500
CS V2R1: Pinging host hosta.test.ibm.com (9.56.99.99)
Ping #1 needs fragmentation at:hoste.test.ibm.com 9.56.22.22
Ping #2 response took 0.002 seconds.
Ping #3 response took 0.001 seconds.
Ping #4 needs fragmentation at: hoste.test.ibm.com 9.56.22.22
```

- IPv4 with the value `ignore` specified for the PMTU parameter, the NOName parameter specified, and fragmentation needed out in the network. The hosts in this IPv4 network do not provide a next-hop MTU value when sending the ICMP error message.

```
ping hosta (count 4 pmtu ignore noname length 2500
CS V2R1: Pinging host hosta.test.ibm.com (9.56.99.99)
Ping #1 needs fragmentation at: (9.56.22.22)
Ping #2 response took 0.002 seconds.
Ping #3 response took 0.001 seconds.
Ping #4 needs fragmentation at: (9.56.33.33)
```

- IPv4 with the Verbose parameter specified.

```
ping hosta (verbose
CS V2R1: Pinging host hosta.test.ibm.com (9.56.99.99)
with 256 bytes of ICMP data
Ping #1 from 9.56.99.99: bytes=264 seq=1 ttl=51 time=1.08 ms
Ping #2 from 9.56.99.99: bytes=264 seq=2 ttl=51 time=1.35 ms
Ping #3 from 9.56.99.99: bytes=264 seq=3 ttl=51 time=1.58 ms

Ping statistics for hosta.test.ibm.com (9.56.99.99)
  Packets: Sent=3, Received=3, Lost=0 (0% loss)
  Approximate round trip times in milliseconds:
  Minimum=1.08 ms, Maximum=1.58 ms, Average=1.34 ms, StdDev=0.26 ms
```

IPv4 with the parameters Verbose, Length, and PMTU specified with the value ignore but with the Ping failures (timeout and fragmentation needed errors)

```
ping hosta (verbose length 2500 PMTU ignore
CS V2R1: Pinging host hosta.test.ibm.com (9.56.99.99)
with 2500 bytes of ICMP data
Ping #1 needs fragmentation at: hoste.test.ibm.com (9.56.22.22)
Ping #2 timed out
Ping #3 timed out

Ping statistics for hosta.test.ibm.com (9.56.99.99)
  Packets: Sent=3, Received=0, Lost=3 (100% loss)
```

- IPv6 with the value ignore specified for the PMTU parameter, the NOName parameter specified, and fragmentation needed out in the network.

```
ping hostipv6 (count 4 pmtu ignore noname length 3000
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
Ping #1 needs fragmentation at: 50c9:c2d4:0:3:9:6b00:111a:250e
  Next-hop MTU size is 1500
Ping #2 response took 0.002 seconds.
Ping #3 response took 0.001 seconds.
Ping #4 needs fragmentation at: 50c9:c2d4:0:3:9:6b00:111a:250e
  Next-hop MTU size is 1500
```

- IPv6 with the value yes specified for the PMTU parameter. Fragmentation needed, first out in the network, and then at the local TCP/IP stack because of Path MTU Discovery.

```
ping hostipv6 (count 4 pmtu yes length 3000
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
Ping #1 needs fragmentation at: hoste.test.ibm.com (50c9:c2d4:0:3:9:6b00:111a:250e)
  Next-hop MTU size is 1500
Ping #2 needs fragmentation at: local.host (50c9:c2d4:0:6:1:6b00:111a:0001)
  Next-hop MTU size is 1500
Ping #3 needs fragmentation at: local.host (50c9:c2d4:0:6:1:6b00:111a:0001)
  Next-hop MTU size is 1500
Ping #4 needs fragmentation at: local.host (50c9:c2d4:0:6:1:6b00:111a:0001)
  Next-hop MTU size is 1500
```

- IPv6 with the Count, Length, and Verbose parameters specified.

```
ping hostipv6 (count 5 length 8944 verbose
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
with 8944 bytes of ICMP data
Ping #1 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=1 hoplim=51 time=1.71 ms
Ping #2 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=2 hoplim=51 time=1.52 ms
Ping #3 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=3 hoplim=51 time=1.78 ms
Ping #4 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=4 hoplim=51 time=1.88 ms
Ping #5 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=5 hoplim=51 time=2.25 ms

Ping statistics for hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
  Packets: Sent=5, Received=5, Lost=0 (0% loss)
  Approximate round trip times in milliseconds:
  Minimum=1.52 ms, Maximum=2.25 ms, Average=1.83 ms, StdDev=0.24 ms
```

IPv6 with the Count, Length, and Verbose parameters specified, but with mixed Ping results (success and failure).

```
ping hostipv6 (count 5 length 8944 verbose
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
with 8944 bytes of ICMP data
Ping #1 timed out
Ping #2 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=2 hoplim=51 time=1.51 ms
Ping #3 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=3 hoplim=51 time=1.68 ms
Ping #4 timed out
Ping #5 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=5 hoplim=51 time=1.64 ms

Ping statistics for hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
  Packets: Sent=5, Received=3, Lost=2 (40% loss)
  Approximate round trip times in milliseconds:
  Minimum=1.51 ms, Maximum=1.68 ms, Average=1.61 ms, StdDev=0.10 ms
```

- IPv6 link-local with scope information.

```
ping fe80::12:1:2%mpc6221
CS V2R1: Pinging host FE80::12:1:2%MPC6221
at IPv6 address fe80::12:1:2
Ping #1 response took 0.028 seconds.
```

Response description:

The Ping command displays one response output line for every echo request packet that is sent. The default response output line displays the number of elapsed seconds for the echo reply that was received and the number of bytes that were sent for the data portion of the echo request packet.

When the Verbose parameter is specified, the following information is displayed:

Echo reply details

Ping #*n* from *address*
Echo reply process counter and IP address of the echo reply sender.

bytes=*nn*
The number of bytes for the ICMP packet (ICMP header and data portions) from the echo reply.

seq=*nn*
ICMP sequence number of the echo reply.

ttl=*nn* (for IPv4)
Time-to-live value for the echo reply.

hoplim=*nn* (for IPv6)
Hop limit value for the echo reply.

time=*nn* ms
Round-trip time (RTT), in milliseconds.

Ping statistics summary

Sent
Total number of echo request packets sent.

Received
Total number of echo reply packets received.

Lost (*n*% loss)
Total number of lost echo packets (echo reply packets that were not received) and the percentage of packets that were lost.

Approximate round trip times (RTT) in milliseconds

Minimum
Minimum RTT value of Ping requests that were sent.

Maximum
Maximum RTT value of Ping requests that were sent.

Average
Average RTT value of Ping requests that were sent.

StdDev
Standard deviation of all RTT values of Ping requests that were sent.

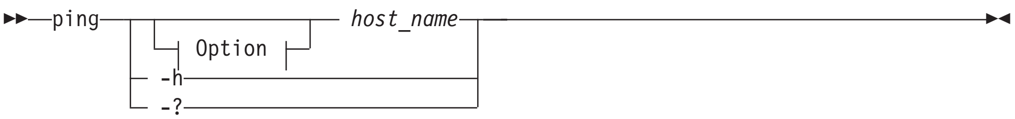
The z/OS UNIX ping command: Send an echo request

The z/OS UNIX **ping** command sends an echo request to a foreign node (remote host) to determine whether the node is accessible.

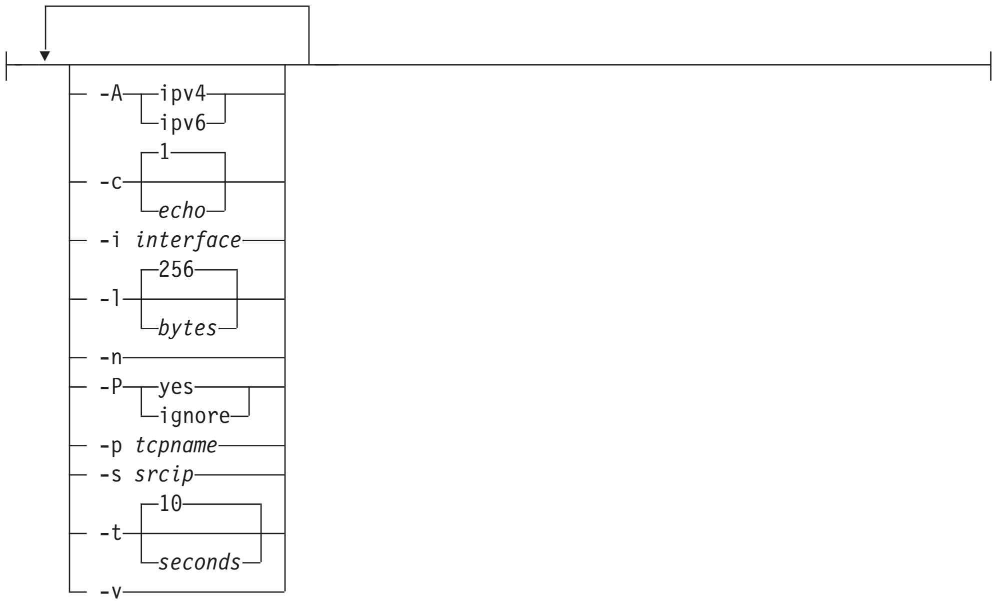
When a response to a Ping command is received, the elapsed time is displayed. The time does not include the time spent communicating between the user and the TCP/IP address space.

Note: ping is a synonym for the **oping** command in the z/OS UNIX shell. The **oping** command syntax is the same as that for the **ping** command.

Format



Option:



Parameters

host_name

Specifies the host to which you want to send the echo request. This must be an IP address or a host name that can be resolved. IPv4-mapped IPv6 addresses are not supported.

If the *host_name* value is specified as a host name (not an IP address), the command invokes the resolver to obtain an IP address for the host name. The command uses the first IP address that is returned by the resolver. Use the **-A** option to specify whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If you do not specify the **-A** option, the -i and **-s** options can also be used to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If neither **-A**, **-i**, or **-s** options are specified, the command does not request a specific type of IP address from the resolver and IPv4 and IPv6 IP addresses can be returned by the resolver.

When using IPv6 link-local addresses, you can provide scope information with the IP address or host name. To specify scope information, add a percent character (%) after the *host_name* value, followed by the scope information (usually an interface name). See the examples that follow for an example of using the command with scope information. For a more complete explanation about the use of scope information, see the support for scope information in the z/OS Communications Server: IPv6 Network and Application Design Guide.

Guidelines:

- When you are running multiple TCP/IP stacks on the same MVS image and the interface name that is used as the scope information has been defined to more than one TCP/IP stack, you must specify the **-p** parameter to ensure that the correct stack is used to send the command's packets.
- Providing scope information on the *host_name* option has the same effect as specifying the local interface using the INTF option, although the **-i** option covers a wider range of situations (scope information applies only to IPv6 link-local addresses). If both methods of providing scope information are used on the same command, the values provided for scope information on the *host_name* option and for the **-i** interface option must represent the same local interface, otherwise the command fails.

-A ipv4 | ipv6

Specifies the IP address type that the Resolver returns when resolving the host name to an IP address. The values for this option are not case sensitive.

ipv6

Specifies that only IPv6 IP addresses are returned from the Resolver when resolving the host name to an IP address.

ipv4

Specifies that only IPv4 IP addresses are returned from the Resolver when resolving the host name to an IP address.

If the **-A** option is not specified see the description of the *host_name* parameter for information on how the *host_name* value is resolved to an IP address.

-c *echo*

Sets the number of echo requests that are sent to the host. If you do not specify the **-c** parameter, the default value 1 is used unless the **-v** parameter is specified. When the **-v** parameter is specified but the **-c** parameter is not specified, the default value is 3. If an *echo* value is not specified, an error occurs. The *echo* value must be in the range 0 - 2^{31} minus 1, which is 2147483647. If *echo* is 0, the Ping command sends echo requests continually. To stop the Ping command, see “Usage” on page 670.

Restriction: If you specify the **-v** parameter, you cannot specify the value 0 for the **-c** parameter.

-h or -question mark (?)
Provides help information about the Ping command. You cannot place the **-h** or -? parameter on the Ping command line with other parameters.

-i *interface*
Specifies the local interface, *interface*, over which the packets are sent. The interface is either a maximum 16-byte name from a LINK or INTERFACE profile statement, or the IP address of a local interface. IPv4-mapped IPv6 addresses are not supported. Local VIPA or LOOPBACK interfaces are not valid.

If the destination host is specified as a host name and the **-A** option is not specified, the address type of the *interface* value is used to determine whether the host name is resolved to an IPv4 or IPv6 IP address.

When this parameter is specified, Ping establishes affinity to either the default TCP/IP stack or the stack specified on the **-p** parameter. The specified interface must be defined to the stack to which Ping establishes affinity. You must also ensure that a route exists to the destination using the specified interface. This can be any kind of route, including a default route. This parameter is independent of the **-s** parameter used as the source IP address in the outbound packets.

Note: As a diagnostics aid in analyzing response times and path availability using a particular route, this parameter routes packets over specified interfaces, regardless of the multipath settings in the IPCONFIG MULTIPATH or IPCONFIG6 MULTIPATH profile statement, by bypassing the outbound path selection algorithm for the packets.

Restriction:

- You cannot specify scope information for the *interface* value.
- To specify an OSM interface for the parameter, the user ID must have RACF authority to use the interface. For more information about OSM interface authorization, see OSM Access Control in z/OS Communications Server: IP Configuration Guide.

Restriction:

-l *bytes*
Sets the number of data bytes for the echo request. If a *bytes* value is not specified, an error occurs. If you do not specify the **-l** parameter, the default value 256 is used. The number of bytes must be in the range 8 - 65 487. A minimum of 8 data bytes is needed for a time stamp value, which Ping uses to correlate echo requests to echo replies.

- For IPv4 destinations, the total length of the outbound echo request packet includes the length of an IPv4 IP header (20 bytes), the length of an ICMP header (8 bytes), and the data length specified by the **-l** parameter. Depending on your TCP/IP stack configuration, the TCP/IP stack might add additional IP header options to the IP header created by the Ping command, before the echo request packet is sent.

- For IPv6 destinations, the total length of the outbound echo request packet includes the length of an IPv6 IP header (40 bytes), the length of an ICMPv6 header (8 bytes), and the data length specified by the **-l** parameter. Depending on your TCP/IP stack configuration, the TCP/IP stack might add additional IPv6 extension headers to the packet created by the Ping command, before the echo request packet is sent.

-n Specifies that the Ping command should not resolve IP addresses to host names for ICMP/ICMPv6 messages received due to Path MTU problems. This parameter is in effect only if the **-P** parameter was also specified, otherwise it is ignored. Specifying this parameter results in the Ping command displaying only the IP address of the host where fragmentation is needed. For example:

```
Ping #n needs fragmentation at: ipaddress
```

-P yes | ignore

This parameter can be used for diagnosing Path Maximum Transmission Unit (MTU) problems in the network. It prevents the outbound echo request packets from being fragmented and specifies what kind of Path MTU Discovery support should be used with the Ping command. For IPv4, Path MTU Discovery support is enabled by specifying the PATHMTUDISCOVERY parameter on the IPCONFIG profile statement. For IPv6, Path MTU Discovery support is enabled by default. The values for this option are not case sensitive.

yes

Specifies that the outbound echo request packets are not fragmented at the local host or in the network and that the MTU value, determined by path MTU discovery for the destination, are used.

- If path MTU discovery has already determined an MTU value for the destination and the length of the Ping echo request packet is larger than this MTU size, then the local TCP/IP stack does not send out the packet. In this case, the Ping command displays one of the local stack's IP addresses as the address of the host where fragmentation is needed and the next-hop MTU value displayed by the Ping command is the current path MTU value to the destination. For Ping commands to IPv4 destinations, the Ping command processing itself does not cause path MTU discovery support to be triggered for the destination. For IPv4, only TCP processing causes path MTU discovery support to be triggered.
- If path MTU discovery is not active, or has not already determined a path MTU value for the destination, and the Ping echo request packet exceeds the configured route MTU selected for this packet, then the local TCP/IP stack does not send out the packet. In this case, the Ping command displays one of the local stack's IP addresses as the address of the host where fragmentation is needed, and the next-hop MTU value that is displayed by the Ping command is that of the route selected for the Ping packet.
- If the Ping request fails because the echo request packet needs to be fragmented at some point in the network, the Ping command displays the IP address where fragmentation needs to occur and displays the next-hop MTU value, if it was provided.

ignore

Specifies that the outbound echo request packets are not fragmented at the local host or in the network, and that any MTU values determined by path MTU discovery for the destination, are ignored.

- If path MTU discovery had determined an MTU value for the destination, and the length of the Ping echo request packet is larger than this MTU size, specifying a value of ignore enables the Ping echo request to be sent out by the local TCP/IP stack, to determine where in the network the original MTU problem occurred. In this case, the Ping command displays the IP address where fragmentation needs to occur and displays the path MTU value, if it was provided.

- If the Ping echo request packet exceeds the configured route MTU selected for this packet, then the local TCP/IP stack does not send out the packet. In this case, the Ping command displays one of the local stack's IP addresses as the address of the host where fragmentation is needed. The next-hop MTU value displayed by the Ping command is that of the route selected for the Ping packet.

If the Ping command receives an ICMP/ICMPv6 error message indicating that an echo request packet needed to be fragmented, the Ping command displays the following output based on this message:

```
Ping #n needs fragmentation at: host_name (ipaddress)
```

If the host name resolution fails, the Ping command displays the following output:

```
Ping #n needs fragmentation at: ipaddress (ipaddress)
```

You can use the **-n** parameter to request that the Ping command display only the host name and its IP address of the host, without resolving it to a host name.

If the host returned the next-hop MTU size in the ICMP/ICMPv6 message, then this MTU size is also displayed: `Next-hop MTU size is nnnnn`

If the MTU size is not displayed, you can use the Length parameter to vary the size of the echo request packet, in order to determine the MTU of the network.

MULTIPATH PERPACKET considerations: When the MULTIPATH PERPACKET option is in effect and equal-cost routes are configured to the Ping destination host, the smallest MTU value of all the equal-cost routes is used as the largest packet size that can be sent, even if some of the equal-cost routes could support a larger packet size.

-p *tcpname*
Specifies the name of the TCP/IP stack to be used.

The *tcpname* is an 8-byte procedure name that is used to start the TCP/IP. When the `S` *`member.identifier`* method of starting TCP/IP is used, the value specified for *identifier* must be used as *tcpname*. When this option is not specified and z/OS UNIX is configured for CINET, the CINET Prerouter selects the TCP/IP stack to which the request is routed.

-s *srcip*
Specifies the source IP address, *srcip*. You must specify this as an IP address and not a host name. IPv4-mapped IPv6 addresses are not supported. On hosts with more than one IP address, you can set the source address to the IP address for another one of the stack's interfaces. This can be a VIPA address.

If the destination host is specified as a host name and the **-A** option is not specified, the address type of the *srcip* value is used to determine whether the host name should be resolved to an IPv4 or IPv6 IP address.

Restriction: You cannot specify scope information for the source IP address.

-t *seconds*
Sets the number of seconds that the Ping command waits for a response. If you

do not specify the **-t** parameter, the default of 10 seconds is used. If the *seconds* value is not specified, an error occurs. The number of seconds specified must be in the range 1 - 100.

-v Provides additional details about the received echo replies and a statistics summary.

If you do not specify the **-v** and **-c** parameters, then the default count of echo requests is 1. If you specify the **-v** parameter without the **-c** parameter, then the default value is 3. If you specify both the **-v** and **-c** parameters, then the number of echo requests is the value specified in the **-c** parameter.

Restriction: If you specify the value 0 for the **-c** parameter, you cannot specify the **-v** parameter.See the examples that follow for the format of Ping command output when the **-v** parameter is specified. See the responses that follow for an explanation of the fields that are used in the verbose information.

Usage

- To stop or interrupt the Ping command, press **Ctrl c**. The interrupt key can be changed by using the OMVS ESCAPE command in the z/OS UNIX shell, or the **stty** command for the RAW shell. For more information about OMVS and **stty** commands,see the z/OS UNIX System Services Command Reference.
- You can place more than one parameter on the Ping command line; however, the **-h** and -? parameters are exceptions and cannot be placed on the Ping command line with other parameters.

Restrictions:

- Ping commands to a remote host might fail if there is a firewall between the two systems, even if the host is reachable using other commands.
- Ping commands to a remote host might be unable to detect path MTU information if there is an IPSec tunnel at any point between the two systems, even if the host is reachable using other commands. For more information about Ping **-P** interactions with IPSec tunnels, see "Resolving TSO PING and z/OS UNIX ping command problems" on page 673.

Examples

- IPv4

```
ping mvs098
CS V2R1: Pinging host mvs098 (9.67.113.11)
Ping #1 response took 0.002 seconds.
```

- IPv6

```
ping linuxipv62.tcp
CS V2R1: Pinging host linuxipv62.tcp.raleigh.ibm.com
at IPv6 address 2001:0db8::1:9:67:114:44
Ping #1 response took 0.002 seconds.
```

- IPv4 with the value `ignore` specified for the **-P** parameter and fragmentation needed out in the network. The hosts in this IPv4 network do not provide a next-hop MTU value when sending the ICMP error message. This example represents a network where there are multiple network paths to the destination.

```
ping -c 4 -l 2500 -P ignore  hosta
CS V2R1: Pinging host hosta.test.ibm.com (9.42.99.99)
Ping #1 needs fragmentation at: hoste.test.ibm.com 9.42.22.22
Ping #2 response took 0.002 seconds.
Ping #3 response took 0.001 seconds.
Ping #4 needs fragmentation at: hoste.test.ibm.com 9.42.22.22
```

- IPv4 with the **-v** parameter specified.

```
ping -v hosta
CS V2R1: Pinging host hosta.test.ibm.com (9.56.99.99)
with 256 bytes of ICMP data
Ping #1 from 9.56.99.99: bytes=264 seq=1 ttl=51 time=1.08 ms
Ping #2 from 9.56.99.99: bytes=264 seq=2 ttl=51 time=1.35 ms
Ping #3 from 9.56.99.99: bytes=264 seq=3 ttl=51 time=1.58 ms

Ping statistics for hosta.test.ibm.com (9.56.99.99)
  Packets: Sent=3, Received=3, Lost=0 (0% loss)
  Approximate round trip times in milliseconds:
  Minimum=1.08 ms, Maximum=1.58 ms, Average=1.34 ms, StdDev=0.26 ms
```

IPv4 with the parameters **-v**, **-l**, and **-P** with value of ignore but with the Ping failures (timeout and fragmentation needed errors)

```
ping -v -l 2500 -P ignore hosta
CS V2R1: Pinging host hosta.test.ibm.com (9.56.99.99)
with 2500 bytes of ICMP data
Ping #1 needs fragmentation at: hoste.test.ibm.com (9.56.22.22)
Ping #2 timed out
Ping #3 timed out

Ping statistics for hosta.test.ibm.com (9.56.99.99)
  Packets: Sent=3, Received=0, Lost=3 (100% loss)
```

- IPv6 with the value ignore specified for the **-P** parameter and fragmentation needed out in the network.

```
ping -c 4 -l 3000 -P ignore -n  hostipv6
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
Ping #1 needs fragmentation at: 50c9:c2d4:0:3:9:6b00:111a:250e
  Next-hop MTU size is 1500
Ping #2 response took 0.002 seconds.
Ping #3 response took 0.001 seconds.
Ping #4 needs fragmentation at: 50c9:c2d4:0:3:9:6b00:111a:250e
  Next-hop MTU size is 1500
```

- IPv6 with the value yes specified for the **-P** parameter and the **-n** parameter specified. Fragmentation needed first out in the network and then at the local TCP/IP stack because of path MTU discovery.

```
ping -c 4 -l 3000 -P yes -n  hostipv6
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
Ping #1 needs fragmentation at: hoste.test.ibm.com (50c9:c2d4:0:3:9:6b00:111a:250e)
  Next-hop MTU size is 1500
Ping #2 needs fragmentation at: local.host (50c9:c2d4:0:6:1:6b00:111a:0001)
  Next-hop MTU size is 1500
Ping #3 needs fragmentation at: local.host (50c9:c2d4:0:6:1:6b00:111a:0001)
  Next-hop MTU size is 1500
Ping #4 needs fragmentation at: local.host (50c9:c2d4:0:6:1:6b00:111a:0001)
  Next-hop MTU size is 1500
```

- IPv6 with the **-c** , **-l** , and **-v** parameters specified.

```
ping -c 5 -l 8944 -v hostipv6
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
with 8944 bytes of ICMP data
Ping #1 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=1 hoplim=51  time=1.71 ms
Ping #2 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=2 hoplim=51  time=1.52 ms
Ping #3 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=3 hoplim=51  time=1.78 ms
Ping #4 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=4 hoplim=51  time=1.88 ms
Ping #5 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=5 hoplim=51  time=2.25 ms

Ping statistics for hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
```

```
  Packets: Sent=5, Received=5, Lost=0 (0% loss)
  Approximate round trip times in milliseconds:
  Minimum=1.52 ms, Maximum=2.25 ms, Average=1.83 ms, StdDev=0.24 ms
```

IPv6 with the **-c** , **-l** , and **-v** parameters specified, but with mixed Ping results (success and failure).

```
ping -c 5 -l 8944 -v hostipv6
CS V2R1: Pinging host hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
with 8944 bytes of ICMP data
Ping #1 timed out
Ping #2 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=2 hoplim=5   1 time=1.51 ms
Ping #3 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=3 hoplim=51  time=1.68 ms
Ping #4 timed out
Ping #5 from 50c9:c2d4:0:5:9:6b00:111a:1: bytes=8952 seq=5 hoplim=5   1 time=1.64 ms

Ping statistics for hostipv6.raleigh.ibm.com
at IPv6 address 50c9:c2d4:0:5:9:6b00:111a:1
  Packets: Sent=5, Received=3, Lost=2 (40% loss)
  Approximate round trip times in milliseconds:
  Minimum=1.51 ms, Maximum=1.68 ms, Average=1.61 ms, StdDev=0.10 ms
```

- IPv6 link-local with scope information.

```
ping fe80::12:1:2%mpc6221
CS V2R1: Pinging host FE80::12:1:2%MPC6221
at IPv6 address fe80::12:1:2
Ping #1 response took 0.001 seconds.
```

Response description:

The Ping command displays one response output line for every echo request packet that is sent. The default response output line displays the number of elapsed seconds for the echo reply that was received and the number of bytes that were sent for the data portion of the echo request packet.

When the **-v** parameter is specified, the following information is displayed:

Echo reply details

Ping *#n* **from** *address*
Processing echo reply counter and IP address of the echo reply sender.

bytes=*nn*
Number of bytes for the ICMP packet (ICMP header and data portions) from the echo reply.

seq=*nn*
ICMP sequence number of the echo reply.

ttl=*nn* **(for IPv4)**
Time-to-live value for the echo reply.

hoplim=*nn* **(for IPv6)**
Hop limit value for the echo reply.

time=*nn* **ms**
Round-trip time (RTT), in milliseconds.

Ping statistics summary

Sent
Total number of echo request packets sent.

Received
Total number of echo reply packets received.

`Lost (`*`n`*`% loss)`
Total number of lost echo packets (echo reply packets not received) and the percentage loss.

`Approximate round trip times (RTT) in milliseconds`

`Minimum`
Minimum RTT value of Ping requests that were sent.

`Maximum`
Maximum RTT value of Ping requests that were sent.

`Average`
Average RTT value of Ping requests that were sent.

`StdDev`
Standard deviation of all RTT values of Ping requests that were sent.

TSO PING and z/OS UNIX ping command return codes

The following list shows the return codes that are generated by the TSO PING and z/OS UNIX **ping** commands:

Code	Description
0	Response
4	No response
8	TCP/IP address space failure (TSO PING only)
12	Socket API failure (z/OS UNIX **ping** only)
100	Incorrect parameter

When a response to a TSO PING or z/OS UNIX **ping** command is received, the elapsed time is displayed. The time does not include the time spent communicating between the user and TCP/IP address space.

Resolving TSO PING and z/OS UNIX ping command problems

A host might fail to respond even after several Ping commands for any of the following reasons:

- The host is not listening to the network.
- The host is inoperative, or some network or gateway leading from the user to the host is inoperative.
- The host is slow because of activity.
- The packet is too large for the host.

The echo request sent by the Ping command does not guarantee delivery. Send more than one Ping command before you assume that a communication failure has occurred.

Use additional Ping commands to communicate with other hosts in the network to determine the condition that is causing the communication failure. However, you need to know the network topology to determine the location of the failure. Issue the Ping commands in the following order until the failure is located.

1. Send a Ping command to your local host.

   A successful Ping command sent to a different host on the same network as the original host suggests that the original host is down, or is not listening to the network.

2. Send a Ping command to a host other than your local host on your local network.
3. Send a Ping command to each intermediate node that leads from your local host to the remote host, starting with the node closest to your local host.

   If you cannot get echoes from any host on that network, the trouble is usually somewhere along the path to the remote hosts. Direct a Ping command to the gateway leading to the network in question. If the Ping command fails, continue to test along the network from the target, until you find the point of the communication breakdown.

The following IPSec tunnel considerations apply when using the Ping command to determine the path MTU information:

- Returned path MTU information displays the tunnel endpoint as the address of the host where fragmentation is needed, not the address of the host within the tunnel where fragmentation was required. If the tunnel originates on the local TCP/IP stack, one of the local stack's IP addresses is displayed.
- The returned next-hop MTU size reflects the size of a packet prior to encapsulation. The size of the IPSec encapsulation overhead has been subtracted from the MTU size.
- In an IPv6 network, a minimum MTU size of 1280 must be supported. If subtracting IPSec encapsulation overhead would cause the MTU size to be less than the minimum MTU value of 1280, the packet is fragmented after encapsulation. This should be rare, occurring only in an IPv6 network with very small MTU values (for example, MTU < 1500).
- For more information about IPSec tunnels, see the IP security information in the z/OS Communications Server: IP Configuration Guide.

Rpcinfo

The TSO RPCINFO and z/OS UNIX **orpcinfo** commands display the servers that are registered and operational with any portmapper or rcpbind servers on your network that use RPC binding protocol Version 2.

The TSO RPCINFO command: Display server information

Use the RPCINFO command to display the servers that are registered and operational with any portmapper or rcpbind servers on your network. The RPCINFO command makes a remote procedure call (RPC) to an RPC server and displays the results.

Tip:

- You can also use z/OS RPCINFO with rpcbind servers that support RPC binding protocol Version 2, such as the z/OS rpcbind server. RPC binding protocol Version 2 is the binding protocol used by the portmapper.
- All IPv4 applications can use RPC binding protocol Version 2 to register with rpcbind servers; some applications might register with rpcbind servers using other binding protocols.
- You can use RPCINFO from another platform to query z/OS rpcbind servers for information about servers that register with a binding protocol other than Version 2.

Restriction:

- IPv6 applications cannot register with rpcbind using RPC binding protocol Version 2.
- The RPCINFO command can query only hosts that resolve to valid IPv4 addresses.
- When an rpcbind server is used in place of portmapper, the RPCINFO command can display information only for servers that registered with an rpcbind server using Version 2 binding protocol.

Format

```
►►──RPCINFO──┬─ -p ──┬──────┬─────────────────────────────────────────────┬──────►◄
             │       └─host─┘                                             │
             ├──┬─ -u ──host ──prognum──┬─────────┬─┬─┬─────────────────┬─┤
             │  │                       └─versnum─┘ │ └─ -n ──portnum ──┘ │
             │  └─ -t ──host ──prognum──┬─────────┬─┘                     │
             │                          └─versnum─┘                       │
             └─ -b ──prognum ──versnum────────────────────────────────────┘
```

Parameters

-p *host*
Queries the portmapper on the specified host and prints a list of all registered RPC programs. If *host* is not specified, the system defaults to the local host name. For more information about how the local host name is defined, see the z/OS Communications Server: IP Configuration Reference.

-u *host prognum versnum*
Sends an RPC call to procedure zero of *prognum* on the specified host using UDP, and reports whether a response is received. The variable *prognum* is the name or number of the RPC program.

-n *portnum*
Specifies the port number to be used for the **-t** and **-u** options in place of the port number that is given by the portmapper.

-t *host prognum versnum*
Sends an RPC call to procedure zero of *prognum* on the specified host using TCP, and reports whether a response is received.

-b *prognum versnum*
Sends an RPC broadcast to procedure zero of the specified *prognum* and *versnum* using UDP, and reports all hosts that respond.

Usage

- The *versnum* value is the version of the *prognum* value; it is not the RPC protocol version number.
- The version number is required for the **-b** parameter. If a version is specified, the RPCINFO command attempts to call that version of the specified program. If a version is not specified, RPCINFO prints error information. For example, if **-u** is specified without a version number, then the RPC program reports the versions of its program that it supports.
- You can also use z/OS RPCINFO with rpcbind servers that support RPC binding protocol Version 2, such as the z/OS rpcbind server. RPC binding protocol Version 2 is the binding protocol used by the portmapper.

- All IPv4 applications can use RPC binding protocol Version 2 to register with an rpcbind server; some applications might register with an rpcbind server using other binding protocols.
- You can use the RPCINFO command from another platform to query z/OS rpcbind servers for information about servers that register with a binding protocol other than Version 2.

Restrictions:

- The RPCINFO **-b** command (broadcast) displays only information within the same network. The broadcast packets do not pass through gateways.
- The RPCINFO **-b** command (broadcast) works only for the UDP transport services and does not find any TCP-based services.
- IPv6 applications cannot use the RPC binding protocol Version 2 to register with rpcbind servers.
- The RPCINFO command can query only hosts that resolve to valid IPv4 addresses.
- When an rpcbind server is used in place of portmapper, the RPCINFO command can display information only for servers that registered with the rpcbind server using the Version 2 binding protocol.

Examples

In the following example, the RPCINFO command is used to query the portmapper on host `mvsx`. The RPCINFO command displays the list of registered programs reported by the portmapper on `mvsx`.

```
READY
rpcinfo -p mvsx
program        vers proto   port
 100000           2   udp    111
 100000           2   tcp    111
 150001           1   udp   1030
 150001           2   udp   1030
 100003           2   tcp   2049
 100003           2   udp   2049
 100003           3   tcp   2049
 100003           3   udp   2049
 100003           4   tcp   2049
 100059           2   udp   1029
 100059           2   tcp   1033
 100044           1   udp   1028
 100044           1   tcp   1032
 100005           1   udp   1027
 100005           1   tcp   1031
 100005           3   tcp   1031
 100005           3   udp   1027
```

The z/OS UNIX orpcinfo/rpcinfo command: Display server information

Use the **`orpcinfo`** command to display the servers that are registered and operational with any portmapper on your network. The **`orpcinfo`** command makes a remote procedure call (RPC) to an RPC server and displays the results.

RPCINFO can query only hosts that resolve to valid IPv4 addresses.

Tips:

- **rpcinfo** is a synonym for the **orpcinfo** command in the z/OS UNIX shell. **rpcinfo** command syntax is the same as that for the **orpcinfo** command.
- You can use the **rpcinfo** command from another platform to query z/OS rpcbind servers for information about servers that register with a binding protocol other than Version 2.
- You can also use the z/OS **rpcinfo** command with rpcbind servers that support RPC binding protocol Version 2, such as the z/OS rpcbind server. RPC binding protocol Version 2 is the binding protocol used by the portmapper.
- All IPv4 applications can use RPC binding protocol Version 2 to register with an rpcbind server; some applications might register with an rpcbind server using other binding protocols.

Format

```
►►──rpcinfo──┬─ - p ─┬──────┬───────────────────────────────────────────┬──►◄
             │       └─host─┘                                           │
             ├─┬─ - u ──host ──prognum──┬─────────┬─┬─┬───────────────┬─┤
             │ │                        └─versnum─┘ │ └─ - n ──portnum─┘ │
             │ └─ - t ──host ──prognum──┬─────────┬─┘                   │
             │                          └─versnum─┘                     │
             ├─ - b ──prognum ──versnum─────────────────────────────────┤
             ├─ - d ──prognum ──versnum─────────────────────────────────┤
             └─?────────────────────────────────────────────────────────┘
```

Parameters

-p *host*
: Queries the portmapper on the specified host and prints a list of all registered RPC programs. If *host* is not specified, the system defaults to the local host name. For more information about how the local host name is defined, see the z/OS Communications Server: IP Configuration Reference.

-u *host prognum versnum*
: Sends an RPC call to procedure zero of *prognum* on the specified host using UDP, and reports whether a response is received. The variable *prognum* is the name or number of the RPC program.

-n *portnum*
: Specifies the port number to be used for the **-t** and **-u** options in place of the port number that is given by the portmapper.

-t *host prognum versnum*
: Sends an RPC call to procedure zero of *prognum* on the specified host using TCP, and reports whether a response is received.

-b *prognum versnum*
: Sends an RPC broadcast to procedure zero of the specified *prognum* and *versnum* using UDP, and reports all hosts that respond.

-d *prognum versnum*
: Deletes the registration for the RPC service specified by the *prognum* and *versnum* values.

-? Specifies the command help.

Tip: The *versnum* value is the version of the *prognum* value; it is not the RPC protocol version number.

Requirement: The version number is required for the **-b** parameter. If a version is specified, the **rpcinfo** command attempts to call that version of the specified program. If a version is not specified, the **rpcinfo** command prints error information. For example, if **-u** is specified without a versnum value, the RPC program reports the versions of its program that it supports.

Restrictions:

- z/OS UNIX **orpcinfo -b** (broadcast) displays information with the same network only. The broadcast packets do not pass through gateways.
- z/OS UNIX **orpcinfo -b** (broadcast) works only for the UDP transport services and does not find any TCP-based services.
- Only a superuser can use the **-d** option.
- IPv6 applications cannot use RPC binding protocol Version 2 to register with rpcbind.
- RPCINFO can query only hosts that resolve to valid IPv4 addresses.
- When rpcbind is used in place of the portmapper, rpcinfo can display information only for servers that registered with rpcbind using the Version 2 binding protocol.

Examples

In the following example, the **orpcinfo** command invokes the nullproc procedure of program 100003 on host mvsx using the TCP protocol. The **orpcinfo** command invokes the nullproc procedure on all versions of 100003 on host mvsx and reports the result.

```
# orpcinfo -t mvsx 100003
 EZA4328I program 100003 version 2 ready and waiting
 EZA4328I program 100003 version 3 ready and waiting
#
```

Provide security product access to rpcbind server registrations:

When you invoke the **rpcinfo** command with the **-d** option, a registration from the portmapper or rpcbind server list of registered applications is deleted. You can define the following security product resource profile in the SERVAUTH class to control the ability of a user to delete registrations from the rpcbind list of registered services:

```
EBZ.RPCBIND.sysname.rpcbindname.REGISTRY
```

See the sample EZARACF member for examples of the security product commands that you can use to create the resource profile.

Rules:

- If the SERVAUTH class is not active or if the security product resource profile is not defined, only UID(0) users can use the **-d** option to delete registrations from the rpcbind list of registered servers.
- In a multilevel secure environment, only users permitted to the resource profile can delete registrations from the rpcbind list of registered servers. If the SERVAUTH class is not active or if the security product resource profile is not defined, no users can delete registrations from the rpcbind list of registered servers.

Restriction: This profile does not control the ability of a user to delete registrations from the portmapper list of registered services.

Provide security product access to rpcbind server target assistance procedures:

When you invoke the **rpcinfo** command with the **-b** option, a target assistance RPC is sent to all portmapper or rpcbind servers that are in its subnet. When the rpcbind server host is multilevel secure, you can define the following security product resource profile in the FACILITY class to control the ability of a user to run this target assistance procedure on your rpcbind host.

```
BPX.POE
```

If the FACILITY class is not active or if the security product resource profile is not defined, all users can use the **-b** option to execute the target assistance RPC.

Restrictions:

- This profile does not control the ability of a user to run target assistance RPCs on a portmapper host.
- This profile applies only to rpcbind servers on multilevel secure hosts.

Traceroute

The TSO TRACERTE and z/OS UNIX traceroute/otracert commands help you debug network problems.

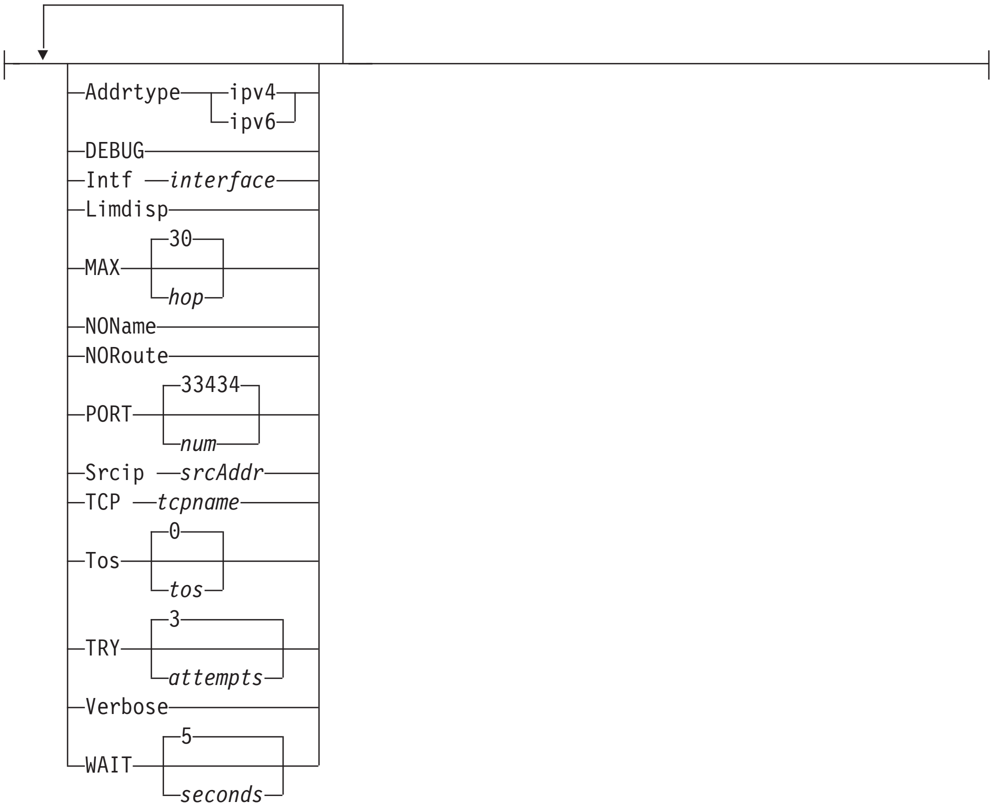
The TSO TRACERTE command: Debug network problems

The TSO TRACERTE command is useful for debugging various network problems. The Tracerte command sends UDP requests with varying TTL (time-to-live) or hop count values and then waits for the routers between the local and remote hosts to send TTL-exceeded messages.

Format

```
►►──TRACERTE───┬─?──────────────────────────────────────────────┬──────────────►◄
               └─host_name──┬────────────┬──┬─────────────────┬─┘
                            └─packetSize─┘  └─(──┤ Options ├──┘
```

Options:



```
  ┌─────────────────────────┐
  ▼                         │
├─┬─Addrtype──┬─ipv4─┬──┬──┴──────────────────────────────────────────┤
  │           └─ipv6─┘  │
  ├─DEBUG───────────────┤
  ├─Intf ──interface────┤
  ├─Limdisp─────────────┤
  │      ┌─30──┐        │
  ├─MAX──┴─hop─┴────────┤
  ├─NOName──────────────┤
  ├─NORoute─────────────┤
  │       ┌─33434─┐     │
  ├─PORT──┴─num───┴─────┤
  ├─Srcip ──srcAddr─────┤
  ├─TCP ──tcpname───────┤
  │      ┌─0───┐        │
  ├─Tos──┴─tos─┴────────┤
  │      ┌─3────────┐   │
  ├─TRY──┴─attempts─┴───┤
  ├─Verbose─────────────┤
  │       ┌─5───────┐   │
  └─WAIT──┴─seconds─┴───┘
```

Note: The minimum abbreviation for each parameter is shown as uppercase letters in the syntax diagram above.

Parameters

? Specifies the command help.

host_name
Specifies the destination host. This must be an IP address, or a host name that can be resolved. IPv4-mapped IPv6 addresses are not supported.

If the *host_name* value is specified as a host name (not an IP address), the command invokes the resolver to obtain an IP address for the *host_name* value. The command uses the first IP address that is returned by the resolver. You can use the ADDRTYPE option to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If you do not specify the ADDRTYPE option, the INTF and SRCIP options can also be used to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If neither ADDRTYPE, INTF, or SRCIP are specified, then the command does not request a specific type of IP address from the resolver; IPv4 and IPv6 IP addresses can be returned by the resolver.

When using IPv6 link-local addresses, you can provide scope information with the IP address or host name. To specify scope information, add a percent character (%) after the *host_name* value, followed by the scope information (usually an interface name). See the examples that follow for an example of using the command with scope information. For a more complete explanation about the use of scope information, see the support for scope information in the z/OS Communications Server: IPv6 Network and Application Design Guide.

Guidelines:

- When you are running multiple TCP/IP stacks on the same MVS image and the interface name used as the scope information has been defined to multiple TCP/IP stacks, you must specify the TCP parameter to ensure that the correct stack is used to send the command's packets.
- Providing scope information on the *host_name* option has the same effect as specifying the local interface using the INTF option, although the INTF option covers a wider range of situations (scope information applies only to IPv6 link-local addresses). If both methods of providing scope information are used on the same command, the values provided for scope information on the *host_name* option and for the INTF interface option must represent the same local interface, otherwise the command fails.

packetSize

Optional parameter that can be used to change the size of a probe packet. The probe size might affect the route of a probe. The value specified is added to the default probe packet size up to a maximum of 65 535 bytes.

For IPv4 destinations, the packet size value must be between 1 and 65 495 bytes. The 65 495 value is the maximum IP packet size (65 535) minus the default probe packet size (40). The default probe packet size includes the IP header, UDP header, and default UDP data.

For IPv6 destinations, the packet size value must be between 1 and 65 515 bytes. The 65 515 value is the maximum UDP data size (65 535) minus the default UDP probe packet size (20). The default probe packet size includes the UDP header, and default UDP data. The IPv6 IP header is added later, before the packet is sent and its size is not included in the packetSize value.

If additional IP headers are dynamically added later to the outbound probe packet then the actual size of the packet is increased.

ADDRTYPE

Specifies the IP address type that the Resolver returns when resolving the host name to an IP address. The values for this option are not case sensitive.

ipv6

Specifies that only IPv6 IP addresses are returned from the Resolver when resolving the host name to an IP address.

ipv4

Specifies that only IPv4 IP addresses are returned from the Resolver when resolving the host name to an IP address.

If the ADDRTYPE option is not specified, see the description of the *host_name* parameter for information on how the *host_name* value is resolved to an IP address.

DEBUG

Specifies that extra messages are to be printed.

INTF *interface*

Specifies the local interface, *interface*, over which the packets are sent. The interface is either a maximum 16-byte name from a LINK or INTERFACE profile statement, or the IP address of a local interface. IPv4-mapped IPv6 addresses are not supported. Local VIPA or LOOPBACK interfaces are not valid.

If the destination host is specified as a host name and the ADDRTYPE option is not specified, the address type of the INTF value is used to determine whether the host name is resolved to an IPv4 or IPv6 IP address.

When this parameter is specified, the Traceroute command establishes affinity to either the default TCP/IP stack or to the stack that is specified on the TCP parameter. The specified interface must be defined to the stack to which the Traceroute command establishes affinity. You must also ensure that a route exists to the destination using the specified interface. This can be any kind of route, including a default route. This parameter is independent of the *SRCIP* parameter used as the source IP address in the outbound packets.

Note: As a diagnostics aid in analyzing response times and path availability using a particular route, this parameter routes packets over specified interfaces regardless of the multipath settings in the IPCONFIG/IPCONFIG6 MULTIPATH profile statements by bypassing the outbound path selection algorithm for the packets.

Restriction:

- You cannot specify scope information for the *interface* value.
- To specify an OSM interface for the parameter, the user ID must have RACF authority to use the interface. For more information about OSM interface authorization, see OSM Access Control in z/OS Communications Server: IP Configuration Guide.

LIMDISP

Displays the hop limit value from each received packet. This value can be used to help detect asymmetric routing.

MAX *hop*

Specifies the maximum time to live (TTL) or hop limit. The range for valid values is 1 - 255. The default is 30.

NONAME

Specifies to print the hop IP address without resolving it to a host name. This address is numeric and saves a name server address-to-name lookup for each gateway on the path.

NOROUTE

Sends information directly to a host in an attached network. If the selected route indicates that the host is not in an adjacent network, an error is returned.

PORT *num*

Specifies the source port number and the starting destination port number. The range for valid values is 2048 - 60 000. The default is 33 434.

For example, in the default case, the source port number is 33 434. The destination port number in the first outbound probe packet is the default port value of 33 434 plus one, or 33 435. The destination port number is incremented by 1 for each subsequent outbound probe packet.

SRCIP *srcAddr*

Specifies the source IP address, *srcAddr*. You must specify this as an IP address and not a host name. IPv4-mapped IPv6 addresses are not supported. On hosts with more than one IP address, you can set the source address to the IP address for another one of the stack's interfaces. This can be a VIPA address.

If the destination host is specified as a host name and the ADDRTYPE option is not specified, the address type of the SRCIP value is used to determine whether the host name should be resolved to an IPv4 or IPv6 IP address.

Restriction: You cannot specify scope information for the source IP address.

TCP *tcpname*
Specifies the name, *tcpname*, of the TCP/IP stack to be used to send the probe packets. The *tcpname* is an 8-byte procedure name that is used to start TCP/IP. When the *member.identifier* method of starting TCP/IP is used, the value specified for *identifier* must be used as *tcpname*. When this option is not specified and z/OS UNIX is configured for CINET, the CINET Prerouter selects the TCP/IP stack to which the request is routed.

TOS *tos*
Specifies the Type of Service value (*tos*) in the probe packets. The range for valid values is 0 - 255. The default is 0. This parameter applies only to IPv4 destinations and is ignored for IPv6 destinations.

TRY *attempts*
Specifies the number of attempts. The range for valid values is 1 - 20. The default is 3.

VERBOSE
Specifies that additional information is to be displayed. The information currently displayed is the number of bytes of the ICMP response and the IP address to which the response was sent.

WAIT *seconds*
Specifies how long to wait for a response. The range for valid values is 1 - 255. The default is 5 seconds.

Results

The Traceroute command displays one line of output for every TTL or hop limit value for which it sent a UDP probe packet. The format of the output is as follows:

```
HOP NAME (IP_ADDRESS) NUM ms !FLAG
```

The values displayed are:

Value	Description
HOP	The hop limit value used in the outbound probe packets.
NAME	If the source IP address in the received Internet Control Message Protocol (ICMP) response can be found in the host site tables, NAME displays the name associated with the source IP address. The host name displayed might include scope information representing the interface over which the ICMP response was received.
IP_ADDRESS	The source IP address from the received ICMP response.
NUM	The elapsed time between when the probe packet was sent out and when the ICMP response to that probe packet was received.
!	An exclamation point without one of the FLAG values below indicates that the received hop limit was less than or equal to 1. Otherwise, an exclamation point is followed by one of the values below.

Value	Description
FLAG	This is an optional field. It is present only if one of the following events occurs. Unless otherwise indicated the flags apply to both IPv4 and IPv6 destinations.

Flag	Indicates
*	No datagram was received before your request timed out. The hop might not respond with ICMP or, the NETACCESS configuration might prohibit the response packets from being received by the command because of the security product user ID associated with the user who invoked the command.
A	Administratively prohibited (IPv6 only).
B	Destination is beyond scope of source address (IPv6 only).
C	Precedence cutoff in effect (IPv4 only).
D	Destination Host unknown (IPv4 only).
F	The packet needs to be fragmented.
H	The destination host is unreachable.
N	The destination network is unreachable (IPv4 only).
P	The destination protocol is unreachable (IPv4 only).
Q	The destination host is reachable, but cannot accept the packet because the queue is full (IPv4 only).
R	No route to destination (IPv6 only).
S	The route supplied for the message was incorrect (IPv4 only).
T	Network unreachable for TOS or host unreachable for TOS (IPv4 only).
U	Address is unreachable (IPv6 only).
V	Host precedence violation (IPv4 only).
X	Communication administratively prohibited by filtering (IPv4 only). Firewall configuration is the most common reason for this code being returned to Traceroute.
num	Unknown ICMP Unreachable code (IPv4 only).

For a list of the ICMP types associated with the preceding Flags, see Appendix E, “ICMP/ICMPv6 types and codes,” on page 1071.

Examples

Note: In these examples, an asterisk (*) represents a lost packet.

- The second hop in this example does not send TTL-exceeded messages.

```
tracerte cyst.watson.ibm.com
CS V2R1: Traceroute to CYST.WATSON.IBM.COM (9.2.91.34)
1 9.67.22.2 (9.67.22.2) 67 ms 53 ms 60 ms
2 * * *
3 9.67.1.5 (9.67.1.5) 119 ms 83 ms 65 ms
4 9.3.8.14 (9.3.8.14) 77 ms 80 ms 87 ms
5 9.158.1.1 (9.158.1.1) 94 ms 89 ms 85 ms
6 9.31.3.1 (9.31.3.1) 189 ms 197 ms *
7 * * 9.31.16.2 (9.31.16.2) 954 ms
8 129.34.31.33 (129.34.31.33) 164 ms 181 ms 216 ms
9 9.2.95.1 (9.2.95.1) 198 ms 182 ms 178 ms
10 9.2.91.34 (9.2.91.34) 178 ms 187 ms *
```

- Sometimes packets are lost (hop 6).

```
tracerte 129.35.130.09
CS V2R1: Traceroute to 129.35.130.09 (129.35.130.9)
1 9.67.22.2 (9.67.22.2) 61 ms 62 ms 56 ms
2 * * *
3 9.67.1.5 (9.67.1.5) 74 ms 73 ms 80 ms
4 9.3.8.1 (9.3.8.1) 182 ms 200 ms 184 ms
5 129.35.208.2 (129.35.208.2) 170 ms 167 ms 163 ms
6 * 129.35.208.2 (129.35.208.2) 192 ms !H 157 ms !H
```

- The network was found, but no host was found. The packet could not route to that network.

```
tracerte 129.45.45.45
CS V2R1: Traceroute to 129.45.45.45 (129.45.45.45)
1 9.67.22.2 (9.67.22.2) 320 ms 56 ms 71 ms
2 * * *
3 9.67.1.5 (9.67.1.5) 67 ms 64 ms 65 ms
4 9.67.1.5 (9.67.1.5) 171 ms !N 68 ms !N 61 ms !N
```

- The Traceroute command uses a domain name server along with the site tables for inverse name resolution. If a host name is found, it is printed along with its IP address.

```
tracerte EVANS
CS V2R1: Traceroute to EVANS (9.67.30.25)
1 BART (9.67.60.85) 20 ms 56 ms 71 ms
2 BUZZ (9.67.60.84) 55 ms 56 ms 54 ms
3 EVANS (9.67.30.25) 67 ms 64 ms 65 ms
```

- Successful Traceroute to an IPv6 destination:

```
tracerte linuxipv62.tcp
CS V2R1: Traceroute to LINUXIPV62.TCP.raleigh.ibm.com
at IPv6 address: 2001:0DB8::1:9:67:114:44
1 2001:0DB8::206:2aff:fe66:c800
   (2001:0DB8::206:2aff:fe66:c800)  2 ms  3 ms *
2 2001:0DB8::1:9:67:114:44
   (2001:0DB8::1:9:67:114:44)  2 ms  2 ms  2 ms
```

- Successful Traceroute to an IPv6 link-local destination:

```
tracerte fe80::12:1:2%mpc6221
CS V2R1: Traceroute to FE80::12:1:2
at IPv6 address: fe80::12:1:2
1 fe80::12:1:2%MPC6221
  (fe80::12:1:2)  62 ms  1 ms  0 ms
```

- Using an unknown IPv6 IP address results in a flag indicating that there is no route to the destination.

```
tracerte 2001:0DB8::1:9:67:114:47
CS V2R1: Traceroute to 2001:0DB8::1:9:67:114:47
at IPv6 address: 2001:0DB8::1:9:67:114:47
1 2001:0DB8::206:2aff:fe66:c800
  (2001:0DB8::206:2aff:fe66:c800)  3 ms !R *  2 ms !R
```

Usage

- To authorize the TSO Traceroute command to use RAW sockets, add the command name, TRACERTE, to the AUTHCMD NAMES section of the member IKJTSOxx of SYS1.PARMLIB. TSO user IDs with UNIX System Services Superuser authority are able to execute the command even without this

SYS1.PARMLIB modification. For other authorization considerations, see MVS-related considerations information in the z/OS Communications Server: IP Configuration Guide.

- The range of port numbers that the Traceroute command uses are typically not valid but you can change the range if the target host is using a nonstandard UDP port.
- To interrupt Traceroute command processing, use the PA1 or ATTN key.

Restrictions:

- If IPv4 tunnels exist on the path to the IPv6 destination host, the IPv4 routers in the tunnel are not counted in the hop count. For a more complete description of tunnels, see the z/OS Communications Server: IPv6 Network and Application Design Guide.
- Traceroute commands to a remote host might be unable to detect TTL or hop limit exceeded messages if there is an IPSec tunnel at any point between the two systems, even if the host is reachable using other commands.

The z/OS UNIX traceroute command: Debug network problems

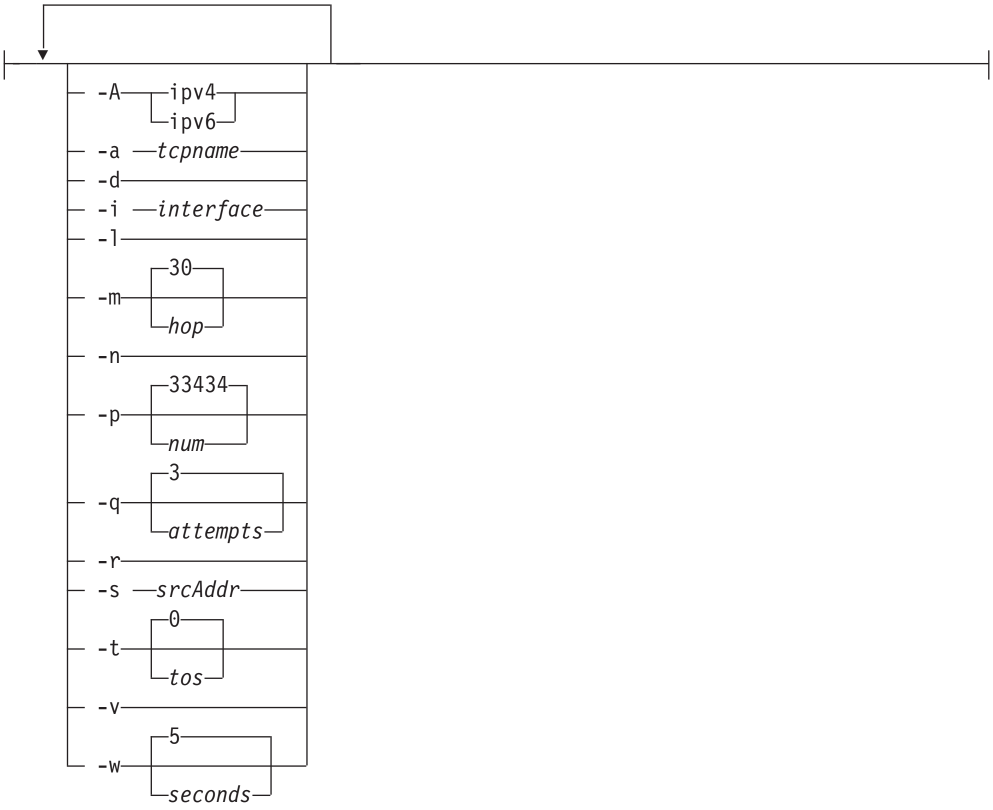
This command is useful for debugging various network problems. This command sends UDP requests with varying TTL (time to live) or hop limit values and then waits for the routers between the local and remote hosts to send time-exceeded messages.

Note: The **otracert** command is a synonym for the **traceroute** command in the z/OS UNIX shell. The **otracert** command syntax is the same as that for the **traceroute** command.

Format

```
►►──traceroute──┬─?───────────────────────────────────────┬──────────────►◄
                └──┬─────────────┬──host_name──┬────────────┬─┘
                   └─┤ Options ├─┘             └─packetSize─┘
```

Options:



Parameters

-? Specifies the command help.

host_name

Specifies the destination host. This must be an IP address or a host name that can be resolved. IPv4-mapped IPv6 addresses are not supported.

If the *host_name* value is specified as a host name (not an IP address), the command invokes the resolver to obtain an IP address for the *host_name* value. The command uses the first IP address that is returned by the resolver. The **-A** option can be used to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If the **-A** option is not specified, the -i and **-s** options can also be used to determine whether the command requests only IPv4 or only IPv6 IP addresses from the resolver. If neither the **-A**, -i, or **-s** options are specified, then the command does not request a specific type of IP address from the resolver, so both IPv4 and IPv6 IP addresses can be returned by the resolver.

When using IPv6 link-local addresses, you can provide scope information with the IP address or host name. To specify scope information, add a percent character (%) after the *host_name* value, followed by the scope information (usually an interface name). An example follows that uses the command with scope information. For a more complete explanation about the use of scope information, see the support for scope information in the z/OS Communications Server: IPv6 Network and Application Design Guide.

Guidelines:

- When you are running multiple TCP/IP stacks on the same MVS image and the interface name used as the scope information has been defined to more than one TCP/IP stack, you must specify the **-a** parameter to ensure that the correct stack is used to send the command's packets.

- Providing scope information on the *host_name* option has the same effect as specifying the local interface using the INTF option, although the **-i** option covers a wider range of situations (scope information applies only to IPv6 link-local addresses). If both methods of providing scope information are used on the same command, the values provided for scope information on the *host_name* option and for the **-i** interface option must represent the same local interface, otherwise the command fails.

`packetSize`
Optional parameter that can be used to change the size of a probe packet. The probe size might affect the route of a probe. The value specified is added to the default probe packet size up to a maximum of 65 535 bytes.

For IPv4 destinations, the packet size value must be between 1 and 65 495 bytes. The 65 495 value is the maximum IP packet size (65 535) minus the default probe packet size (40). The default probe packet size includes the IP header, UDP header, and default UDP data.

For IPv6 destinations, the packet size value must be between 1 and 65 515 bytes. The 65 515 value is the maximum UDP data size (65 535) minus the default UDP probe packet size (20). The default probe packet size includes the UDP header, and default UDP data. The IPv6 IP header is added later, before the packet is sent and its size is not included in the packetSize value.

If additional IP headers are dynamically added later to the outbound probe packet then the actual size of the packet is increased.

-A Specifies the IP address type that the Resolver returns when resolving the host name to an IP address. The values for this option are not case sensitive.

ipv6
Specifies that only IPv6 IP addresses are returned from the Resolver when resolving the host name to an IP address.

ipv4
Specifies that only IPv4 IP addresses are returned from the Resolver when resolving the host name to an IP address.

If the **-A** option is not specified, see the description of the *host_name* parameter for information on how the *host_name* value is resolved to an IP address.

-a *tcpname*
Specifies the name of the TCP/IP stack to be used to send the probe packets. The *tcpname* is an 8-byte procedure name that is used to start TCP/IP. When the S *member.identifier* method of starting TCP/IP is used, the value specified for *identifier* must be used as the *tcpname* value.

When the **-a** option is not specified and z/OS UNIX is configured for CINET, the CINET Prerouter selects the TCP/IP stack to which the request is routed.

-d Specifies that extra messages and other debugging information are to be displayed.

-i *interface*
Specifies the local interface over which the packets is sent. The interface is either a maximum 16-byte name from a LINK or INTERFACE profile statement, or it is the IP address of the local interface. IPv4-mapped IPv6 addresses are not supported. Local VIPA or LOOPBACK interfaces are not valid.

If the destination host is specified as a host name and the **-A** option is not specified, the address type of the -i value is used to determine whether the host name is resolved to an IPv4 or IPv6 IP address.

When this parameter is specified, the command establishes affinity to either the default TCP/IP stack or the stack specified on the **-a** parameter. The specified interface must be defined to the stack to which the command establishes affinity. You must also ensure that a route exists to the destination using the specified interface. This can be any kind of route, including a default route. This parameter is independent of the **-s** parameter used as the source IP address in the outbound packets.

Note: As a diagnostics aid in analyzing response times and path availability using a particular route, this parameter routes packets over specified interfaces regardless of the multipath settings in the IPCONFIG/IPCONFIG6 MULTIPATH profile statement by bypassing the outbound path selection algorithm for the packets.

Restriction:

- You cannot specify scope information for the *interface* value.
- To specify an OSM interface for the parameter, the user ID must have RACF authority to use the interface. For more information about OSM interface authorization, see OSM Access Control in z/OS Communications Server: IP Configuration Guide.

-l

Displays the time-to-live or hop limit value from each received packet. This value can be used to help detect asymmetric routing.

-m *hop*

Specifies the maximum time to live or hop limit. The range for valid values is 1 - 255. The default is 30.

-n

Specifies to print the hop IP address without resolving it to a host name. This address is numeric and saves a name server address-to-name lookup for each gateway on the path.

-p *num*

Specifies the starting destination port number. This parameter does not affect the value of the source port number used. The range of valid values is 2048 - 60 000. The default is 33 434.

For example, in the default case, the destination port number in the first outbound probe packet is the default port value of 33 434 plus 1, or 33 435. The destination port number is incremented by 1 for each subsequent outbound probe packet.

-q *attempts*

Specifies the number of times that a probe is sent with the same time-to-live/hop limit value. This number reflects the total probe transmission (success or failure) per time-to-live/hop limit increment. The range is 1 - 20. The default is 3.

-r

Sends information directly to a host in an attached network. If the selected route indicates that the host is not in an adjacent network, an error is returned.

-s *scrAddr*

Specifies the source IP address. You must specify this address as an IP number and not a host name. IPv4-mapped IPv6 addresses are not supported. On hosts with more than one IP address, you can set the source address to the IP address for another one of the stack's interfaces. This can be a VIPA address.

If the destination host is specified as a host name and the **-A** option is not specified, the address type of the **-s** value is used to determine whether the host name should be resolved to an IPv4 or IPv6 IP address.

Restriction: You cannot specify scope information for the source IP address.

-t *tos*
Specifies the Type of Service value (*tos*) in the probe packets. The range for valid values is 0 - 255. The default is 0. This parameter applies only to IPv4 destinations and is ignored for IPv6 destinations.

-v Specifies that additional information is to be displayed. The information currently displayed is the number of bytes of the ICMP response and the IP address to which the response was sent.

-w *seconds*
Specifies how long to wait for a response. The range for valid values is 1 - 255. The default is 5 seconds.

Results

The traceroute command displays one line of output for every TTL value for which it sent a UDP probe packet. The format of the output is as follows:

```
HOP NAME (IP_ADDRESS) NUM ms FLAG
```

The values displayed are:

Value	Description
HOP	The hop limit value used in the outbound probe packets.
NAME	If the source IP address in the received Internet Control Message Protocol (ICMP) response can be found in the host site tables, NAME displays the name associated with the source IP address. The host name displayed might include scope information representing the interface over which the ICMP response was received.
IP_ADDRESS	The source IP address from the received ICMP response.
!	An exclamation point without one of the FLAG values below, indicates that the received hop limit was less than or equal to 1. Otherwise, an exclamation point is followed by one of the values below.
NUM	The elapsed time between when the probe packet was sent out and when the ICMP response to that probe packet was received.

Value	Description
FLAG	This is an optional field. It is present only if one of the following events occurs. Unless otherwise indicated the flags apply to both IPv4 and IPv6 destinations.

Flag	Indicates
*	No datagram was received before your request timed out. The hop might not respond with ICMP or, the NETACCESS configuration might prohibit the response packets from being received by the command because of the security product user ID associated with the user who invoked the command.
A	Administratively prohibited (IPv6 only).
B	Destination is beyond scope of source address (IPv6 only).
C	Precedence cutoff in effect (IPv4 only).
D	Destination Host unknown (IPv4 only).
F	The packet needs to be fragmented.
H	The destination host is unreachable.
N	The destination network is unreachable (IPv4 only).
P	The destination protocol is unreachable (IPv4 only).
Q	The destination host is reachable, but cannot accept the packet because the queue is full (IPv4 only).
R	No route to destination (IPv6 only).
S	The route supplied for the message was incorrect (IPv4 only).
T	Network unreachable for TOS or host unreachable for TOS (IPv4 only).
U	Address is unreachable (IPv6 only).
V	Host precedence violation (IPv4 only).
X	Communication administratively prohibited by filtering (IPv4 only). Firewall configuration is the most common reason for this code being returned to Traceroute.
num	Unknown ICMP Unreachable code (IPv4 only).

For a list of the ICMP types associated with the preceding Flags, see Appendix E, "ICMP/ICMPv6 types and codes," on page 1071.

Examples

In these examples, an asterisk (*) represents a lost packet.

- The second hop in this example does not send TTL-exceeded messages.

```
traceroute cyst.watson.ibm.com
CS V2R1: Traceroute to CYST.WATSON.IBM.COM (9.2.91.34)
Enter ESC character plus C or c to interrupt
1 9.67.22.2 (9.67.22.2) 67 ms 53 ms 60 ms
2 * * *
3 9.67.1.5 (9.67.1.5) 119 ms 83 ms 65 ms
4 9.3.8.14 (9.3.8.14) 77 ms 80 ms 87 ms
5 9.158.1.1 (9.158.1.1) 94 ms 89 ms 85 ms
6 9.31.3.1 (9.31.3.1) 189 ms 197 ms *
7 * * 9.31.16.2 (9.31.16.2) 954 ms
8 129.34.31.33 (129.34.31.33) 164 ms 181 ms 216 ms
9 9.2.95.1 (9.2.95.1) 198 ms 182 ms 178 ms
10 9.2.91.34 (9.2.91.34) 178 ms 187 ms *
```

- Sometimes packets are lost (hop 6).

```
traceroute 129.35.130.09
CS V2R1: Traceroute to 129.35.130.09 (129.35.130.9)
Enter ESC character plus C or c to interrupt
1 9.67.22.2 (9.67.22.2) 61 ms 62 ms 56 ms
2 * * *
9.67.1.5 (9.67.1.5) 74 ms 73 ms 80 ms
4 9.3.8.1 (9.3.8.1) 182 ms 200 ms 184 ms
5 129.35.208.2 (129.35.208.2) 170 ms 167 ms 163 ms
6 * 129.35.208.2 (129.35.208.2) 192 ms !H 157 ms !H
```

- The network was found, but no host was found. The packet could not route to that network.

```
traceroute 129.45.45.45
CS V2R1: Traceroute to 129.45.45.45 (129.45.45.45)
Enter ESC character plus C or c to interrupt
1 9.67.22.2 (9.67.22.2) 320 ms 56 ms 71 ms
2 * * *
3 9.67.1.5 (9.67.1.5) 67 ms 64 ms 65 ms
4 9.67.1.5 (9.67.1.5) 171 ms !N 68 ms !N 61 ms !N
```

- z/OS UNIX traceroute uses a domain name server along with the site tables for inverse name resolution. If a host name is found, it is printed along with its IP address.

```
traceroute EVANS
CS V2R1: Traceroute to EVANS (129.45.45.45)
Enter ESC character plus C or c to interrupt
1 BART (9.67.60.85) 20 ms 56 ms 71 ms
2 BUZZ (9.67.60.84) 55 ms 56 ms 54 ms
3 EVANS (9.67.30.25) 67 ms 64 ms 65 ms
```

- Successful traceroute to an IPv6 destination.

```
traceroute linuxipv62.tcp
CS V2R1: Traceroute to linuxipv62.tcp.raleigh.ibm.com
at IPv6 address: 2001:0DB8::1:9:67:114:44
Enter ESC character plus C or c to interrupt
1 2001:0DB8::206:2aff:fe66:c800
  (2001:0DB8::206:2aff:fe66:c800)  2 ms  3 ms *
2 2001:0DB8::1:9:67:114:44
  (2001:0DB8::1:9:67:114:44)  2 ms  2 ms  2 ms
```

- Successful traceroute to an IPv6 link-local destination.

```
traceroute fe80::12:1:2%mpc6221
CS V2R1: Traceroute to fe80::12:1:2
at IPv6 address: fe80::12:1:2
Enter ESC character plus C or c to interrupt
1 fe80::12:1:2%MPC6221
  (fe80::12:1:2)  1 ms  2 ms  1 ms
```

- Using an unknown IPv6 IP address results in a flag indicating that there is no route to the destination.

```
traceroute 2001:0DB8::1:9:67:114:47
CS V2R1: Traceroute to 2001:0DB8::1:9:67:114:47
at IPv6 address: 2001:0DB8::1:9:67:114:47
Enter ESC character plus C or c to interrupt
1 2001:0DB8::206:2aff:fe66:c800
  (2001:0DB8::206:2aff:fe66:c800)  3 ms !R *  2 ms !R
```

Usage

- The range of port numbers the traceroute command uses is normally not valid but can be changed if the target host is using nonstandard UDP port.
- To interrupt traceroute command processing, enter the ESC character plus the letter C or c. For example, if the ESC character for the UNIX shell is $, enter $c or $C.

Restrictions:

- If IPv4 tunnels exist on the path to the IPv6 destination host, the IPv4 routers in the tunnel are not counted in the hop count. For a more complete description of tunnels, see the z/OS Communications Server: IPv6 Network and Application Design Guide.
- Traceroute commands to a remote host might be unable to detect TTL or hop limit exceeded messages if there is an IPSec tunnel at any point between the two systems, even if the host is reachable using other commands.

Chapter 4. Managing network security

This topic describes the following commands that are used to manage network security:

- "ipsec command"
- "nssctl command" on page 808
- "certbundle command" on page 814

For additional information that is useful for managing security, see the following topics:

- "MODIFY command: Network security services server" on page 190
- "MODIFY command: IKE server" on page 187
- "MODIFY command: Defense Manager daemon" on page 181
- "The z/OS UNIX pasearch command: Display policies" on page 819
- "Netstat CONFIG/-f report" on page 383 shows the IPSECURITY setting defined on the IPCONFIG and IPCONFIG6 TCP/IP statements
- "Netstat DEvlinks/-d report" on page 425 shows the IPSec security class value as specified on the LINK or INTERFACE TCP/IP statements

ipsec command

The z/OS UNIX **ipsec** command displays and modifies IP security information for a local TCP/IP stack and the IKE daemon or for a network security services (NSS) IPSec client that uses the IPSec network management service of the local NSS server. You can configure a TCP/IP stack as an NSS IPSec client by adding a NssStackConfig statement to the configuration file of the stack's IKE daemon. See z/OS Communications Server: IP Configuration Guide for details. The NSS client can reside on the local z/OS system or on a different z/OS system.

You can also use the **ipsec** command to display, add, and manage defensive filters in the TCP/IP stack and the Defense Manager daemon (DMD). An external security information and event manager typically adds defensive filters in response to a detected intrusion. See defensive filtering information in z/OS Communications Server: IP Configuration Guide for more information about the defensive filters and the DMD. The **ipsec** command displays and modifies defensive filter information for a local TCP/IP stack or for all stacks on a local z/OS image for which the DMD is managing defensive filters.

IP security is implemented through a set of entities that is shared between the TCP/IP stack and the IKE daemon. For a description of the terms and concepts that are used, see IP security information in the z/OS Communications Server: IP Configuration Guide.

You can use the **ipsec** command for the following IP security management activities:

- Display the default or current filter rules and change the filter rule set that the stack is using
- Activate, deactivate, display, and refresh manual and dynamic IPSec tunnels
- Deactivate, display, and refresh IKE tunnels

- Display stack interfaces, including their security class and DVIPA status
- For a particular type of data traffic between two specific endpoints, display which filter rules apply, including both defensive filters and IP security filters
- Display information about the active NSS IPSec client configuration
- Display information maintained by the NSS server for each NSS IPSec client

The **ipsec** command is also used to display and manage defensive filters on the local host system.

Restriction: You cannot display and manage defensive filters for an NSS IPSec client.

You can use the **ipsec** command for the following defensive filter management activities:

- Add a defensive filter to a specific stack or globally to all eligible stacks. (An eligible stack is a stack on the local z/OS image that is enabled for IP security and that is included in the Defense Manager daemon (DMD) configuration file and has the mode active or simulate.)
- Display defensive filters that are installed in a specific stack.
- Display global defensive filters.
- Delete a defensive filter from a specific stack or globally from all eligible stacks.
- Update a defensive filter that is installed in a specific stack or globally in all eligible stacks.
- For a particular type of data traffic between two specific endpoints, display which filter rules apply, including both defensive filters and IP security filters

Tips:

- Use Secure Shell (SSH) from remote machines to issue secure **ipsec** commands.
- The DMD supports up to 10 concurrent **ipsec** command connections. Automated solutions should issue the **ipsec** commands serially to ensure that each **ipsec** command invocation can open a successful connection to the Defense Manager daemon (DMD).

As new functionality is added to the z/OS Communications Server, the **ipsec** command input options and display output might change. Programs that post process the output of the **ipsec** command might be affected by the introduction of z/OS Communications Server maintenance or the installation of a later release. The z/OS Summary of Message and Interface Changes includes information about changes to **ipsec** command reports.

ipsec command security

The **ipsec** command is an APF-authorized application. Users of the **ipsec** command must also be authorized through the security access facility (SAF). This information assumes that the SAF is RACF. Authorization is managed with the SERVAUTH profile and is described in "ipsec command SERVAUTH profile" on page 697. For the **ipsec -f default** and **ipsec -f reload** command, file system access is also required. You do not need root authority to use the **ipsec** command, but for filter rule set control on a local stack, the administrator must provide you with some file access capability. For more information about granting group access control, see "Group access control for local host stacks" on page 699.

ipsec command SERVAUTH profile

Security product authorization (for example, RACF) is required to use the **ipsec** command. You must define a profile in the SERVAUTH class to enable control over the **ipsec** command function. You can define separate profiles during installation to control access to different aspects of the **ipsec** command. The format of the profile when accessing a local stack is as follows:

```
EZB.IPSECCMD.sysname.stackname.command_type
```

Where:

sysname
: The name of the system on which the **ipsec** command is allowed to run.

stackname
: The *tcpprocname* value of the local TCP/IP stack for which the **ipsec** command is authorized. Specify the *stackname* value DMD_GLOBAL to authorize the use of the global defensive filter option (**-G**). The wildcard value asterisk (*) authorizes the use of the global defensive filter option and authorizes all stacks.

command_type
: The **ipsec** command type; either DISPLAY or CONTROL

Table 18. ipsec command SERVAUTH class resource names

Resource names in SERVAUTH class	ipsec options
EZB.IPSECCMD.*sysname*.*stackname*.*	All **ipsec** options
EZB.IPSECCMD.*sysname*.*stackname*.DISPLAY	**-f display** **-F display** **-m display** **-k display** **-y display** **-t** **-i** **-o**
EZB.IPSECCMD.*sysname*.*stackname*.CONTROL	**-f default** **-f reload** **-F add** **-F delete** **-F update** **-m activate** **-m deactivate** **-k deactivate** **-k refresh** **-y activate** **-y deactivate** **-y refresh**
EZB.IPSECCMD.*sysname*.DMD_GLOBAL.DISPLAY	**-F display -G**
EZB.IPSECCMD.*sysname*.DMD_GLOBAL.CONTROL	**-F add -G** **-F delete -G** **-F update -G**

Table 18. ipsec command SERVAUTH class resource names (continued)

Resource names in SERVAUTH class	ipsec options
EZB.IPSECCMD.*sysname*.*stackname*.CONTROL (for each stack to which the global command applies)	**-F add -G** **-F delete -G** **-F update -G**

When accessing a remote stack using the NSS server, the following format applies:

```
EZB.NETMGMT.sysname.clientname.IPSEC.command_type
```

Where:

sysname
The system name on which the **ipsec** command is allowed to run.

clientname
The name of an NSS client.

command_type
The **ipsec** command type; either DISPLAY or CONTROL.

Requirement: You must define these profiles on the system where the NSS server and the **ipsec** command are running

Resource names in SERVAUTH class	ipsec options
EZB.NETMGMT.*sysname*.*clientname*.IPSEC.*	All ipsec options
EZB.NETMGMT.*sysname*.*clientname*.IPSEC.DISPLAY	**-f display** **-m display** **-k display** **-y display** **-t** **-i** **-o**
EZB.NETMGMT.*sysname*.*clientname*.IPSEC.CONTROL	**-f default** **-f reload** **-m activate** **-m deactivate** **-k deactivate** **-k refresh** **-y activate** **-y deactivate** **-y refresh**

Restriction: You cannot display and manage defensive filters for an NSS client that is managed by the NSS server.

Use the following format when querying IKED for NSS configuration information using the **ipsec -w** command:

```
EZB.NETMGMT.sysname.sysname.IKED.DISPLAY
```

Where:

sysname
The name of the system on which the **ipsec** command is allowed to run.

Requirement: This profile must be defined on the system where IKED and the **ipsec** command are running.

The format of the profile when accessing the NSS server using the **ipsec -x** command is:

```
EZB.NETMGMT.sysname.sysname.NSS.DISPLAY
```

Where:

sysname
The name of the system on which the **ipsec** command is allowed to run.

Requirement: This profile must be defined on the system where the NSS server and the **ipsec** command are running.

If the security product is RACF, you can use the control statements in the sample JCL job that is provided in SEZAINST(EZARACF) to define these authorizations. If the SERVAUTH class is not active or if a matching SERVAUTH policy is not found, the **ipsec** request is rejected.

Tip: Authorization is not required for the help option (**ipsec -?**).

Group access control for local host stacks

A user does not require root authority to use the **ipsec** command, but to avoid erroneous or malicious manipulations of files that are used by the **ipsec** command, the administrator must perform the following steps to require group access control. Group access control is required for the following commands:

- To change filter sets in the stack on the local system (**ipsec -f default** or **reload**), the **ipsec** command creates or deletes a specific marker file that the stack accesses.
- To activate, delete, display, or refresh tunnels (any **ipsec -k** or **ipsec -y**), the **ipsec** command uses an AF_UNIX socket file to communicate with the IKE daemon.

Steps for creating group access control over the path for the ipsec command:

1. Create a supplementary RACF group, assign it a group ID (*gid*), and ensure that the primary administrator is a member of the group.

   ```
   ADDGROUP IKE OMVS(GID(931))
   CONNECT user-special GROUP(IKE) UACC(READ)
   ```

2. Issue the following UNIX System Services commands to set file management at the group level. The path for the **ipsec** command files is /var/ike.

   ```
   chgrp IKE /var/ike
   chmod 2770 /var/ike
   ```

3. Use RACF commands to control which users can manipulate files.

   ```
   CONNECT USER5 GROUP(IKE) UACC(READ)
   REMOVE USER5 GROUP(IKE)
   ```

The z/OS UNIX ipsec command syntax

Use the z/OS UNIX **ipsec** command to display and modify IP security information and defensive filter information on the host z/OS system. With the **-z** option or the **-x** primary option specified, the **ipsec** command displays and modifies IP security information for NSS IPSec clients using the IPSec network management service.

Restriction: When you use the **ipsec** command to interface with the NSS IPSec network management service, you must issue the **ipsec** command on the same host z/OS system on which the NSS server is running.

To display and modify IP security information, the **ipsec** command interacts with both the IKE daemon and a TCP/IP communications stack. One or more stacks can be running concurrently on the host z/OS system. While there is at most one IKE daemon, its data is managed on a per stack basis. The **ipsec** command reports IKED NSS IPSec client information using the **-w** primary option for multiple stacks. It reports NSS server information using the **-x** primary option for multiple NSS IPSec clients. For the other **ipsec** command primary options, the **ipsec** command is always specified for a single stack (using the **-p** option) or NSS IPSec client (using the **-z** option). If the **-p** option and the **-z** option are not specified, the command is directed to the default stack on the local system. The default stack refers to the default TCP/IP address space that is specified on the TCPIPJOBNAME statement in the resolver configuration data set.

To display and modify defensive filter information, the **ipsec** command interacts with both the Defense Manager daemon (DMD) and a TCP/IP communications stack. One or more stacks can be running concurrently on the host z/OS system. Only one DMD can be running on the system. Direct the **ipsec** command **-F** primary option to the DMD by specifying the **-G** (global scope) option. If the **-G** option is not specified, the **ipsec** command **-F** option is directed to a single stack. This can be the stack that is specified with the **-p** option or the default stack. The default stack is the default TCP/IP address space that is specified on the TCPIPJOBNAME statement in the resolver configuration data set.

Restriction: You cannot display and manage defensive filters for an NSS IPSec client using the **-z** option.

The actual configuration of IP security entities is managed through Policy Agent policy file specifications. In the policy file definition, network resources and collections of network resources receive names that assist in the management process. Use **ipsec** command options **-n**, **-g**, and **-l** to identify resources by their policy specification name.

Defensive filters are not configured in Policy Agent policy files. You can add defensive filters to the TCP/IP stack in response to a detected intrusion with the **ipsec** command defensive filter add command. The defensive filter's name is assigned on the add action. Use the **ipsec** command option **-N** to identify a defensive filter by its name.

Rule: All policy names and defensive filter names are case sensitive.

Tip: Use spaces or commas as valid delimiters to separate **ipsec** command parameter values.

Additionally, as tunnels are initiated and established, they also receive a system-assigned name, known as a tunnel ID. System-assigned tunnel IDs take the form of an integer prefixed with a single letter that identifies the tunnel type. The prefix can be M (manual), K (Internet Key Exchange), or Y (dynamic). The integer is based on a 32-bit counter that is incremented at each assignment and wraps at 4,294,967,295. Remember that tunnel IDs are arbitrary and transitory strings. Manual tunnel IDs are assigned when a manual tunnel is installed in the stack by the Policy Agent. A change in the manual tunnel policy definition results in assignment of a new manual tunnel ID. Dynamic and IKE tunnel IDs are assigned when a tunnel is established. They remain consistent for the life of the stack and the life of the IKE daemon. Use the **-a** option to identify resources by their tunnel ID.

In addition to the brief help (**ipsec -?**), a man page describes the command syntax and options in detail (**man ipsec**). The **ipsec** command options are discussed in the following sections.

Format

```
►►──ipsec──┤ Primary Option ├──┤ Global Option ├────────────────────────────►◄
```

Primary Option:

```
├──┬─ -f─┤ IP Filter Option ├──┤ Stackname Option ├────────────┬──────────────┤
   ├─ -F─┤ Defensive Filter Option ├──┤ Target Option ├────────┤
   ├─ -m─┤ Manual Tunnel Option ├──┤ Stackname Option ├────────┤
   ├─ -k─┤ IKE Tunnel Option ├──┤ Stackname Option ├───────────┤
   ├─ -y─┤ Dynamic Tunnel Option ├──┤ Stackname Option ├───────┤
   ├─ -i─┤ Interface Option ├──┤ Stackname Option ├────────────┤
   ├─ -t─┤ IP Traffic Test Option ├──┤ Stackname Option ├──────┤
   ├─ -o─┤ NATT Port Translation Option ├──┤ Stackname Option ├─┤
   ├─ -w─┤ IKED Network Security Option ├──────────────────────┤
   ├─ -x─┤ Network Security Server Option ├──┬──────────────────┬┤
   │                                         └─ -znsclienttname─┘│
   └─ -?─────────────────────────────────────────────────────────┘
```

Global Option:

```
├── -d──┬─ 3──────────┬──────────────────────────────────────────────────────┤
        └─ debuglevel─┘
```

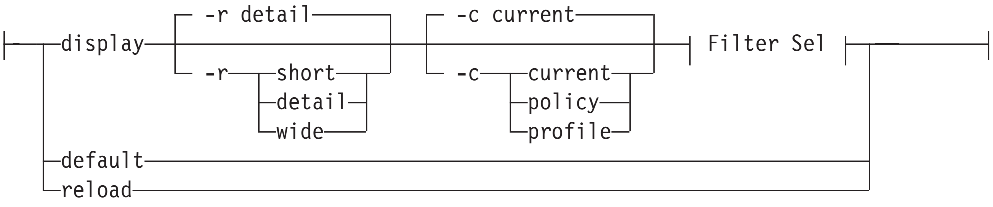
Stackname Option:

```
├──┬─ -p stackname────┬──────────────────────────────────────────────────────┤
   └─ -z nsclientname─┘
```

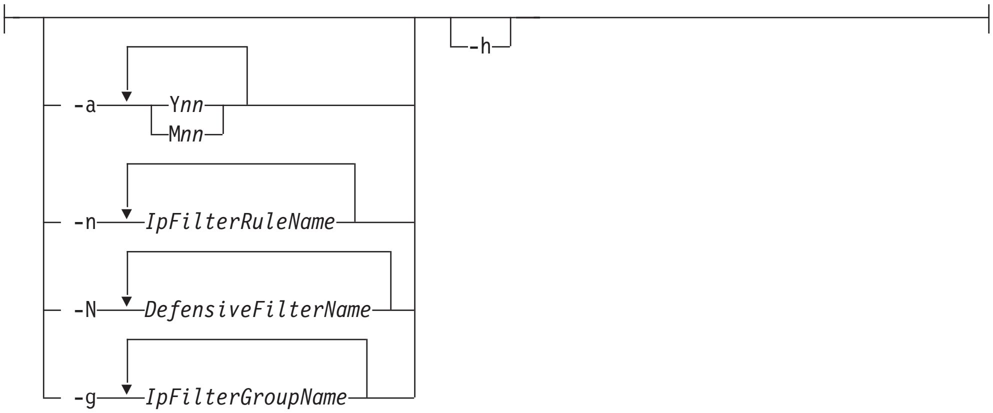
Target Option:

```
├──┬─ -p stackname─┬─────────────────────────────────────────────────────────┤
   └─ -G───────────┘
```

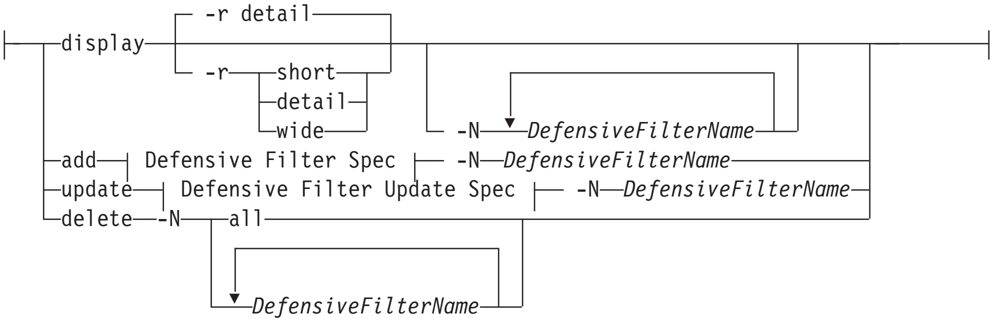
IP Filter Option:



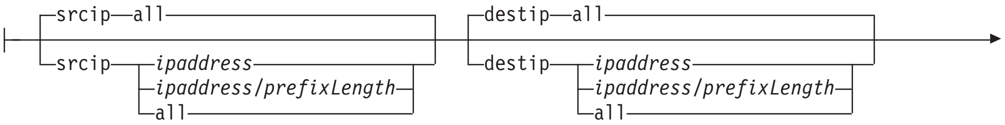
Filter Selection:



Defensive Filter Option:



Defensive Filter Specification:



```
   ┌─prot──all────────────────────────────────────────┐ ┌─dir──inbound──────────┐
►──┼──────────────────────────────────────────────────┼─┼───────────────────────┼──────────►
   └─prot──────┬─┬─tcp─┬──┤ PortSpecification ├───────┤ └─dir───┬─outbound─┬───┘
               │ └─6───┘                              │         └─inbound──┘
               ├─┬─udp─┬──┤ PortSpecification ├───────┤
               │ └─17──┘                              │
               ├─┬─icmp─┬──┤ IcmpSpecification ├──────┤
               │ └─1────┘                             │
               ├─┬─icmpv6─┬──┤ IcmpSpecification ├────┤
               │ └─58─────┘                           │
               ├─igmp─────────────────────────────────┤
               ├─ospf─────────────────────────────────┤
               ├─opaque───────────────────────────────┤
               ├─n────────────────────────────────────┤
               └─all──────────────────────────────────┘
```

```
   ┌─routing──local───────────────────────────────────────┐ ┌─mode──block──────────┐
►──┼──────────────────────────────────────────────────────┼─┼──────────────────────┼──────────►
   └─routing──┬─local──────────────────────────────────┤ └─mode───┬─block────┬─┘
              ├─routed─┬─┤ FragmentSpecification ├─┤  │          └─simulate─┘
              └─either─┘                            └──┘
```

```
   ┌─log──yes──────┐
►──┼───────────────┼──────────────────────────────────────────────────────────────────────►
   └─log───┬─yes─┬─┘
           └─no──┘
```

```
   ┌─loglimit──value_of_DMD_configuration_DefaultLogLimit_parameter─┐
►──┼────────────────────────────────────────────────────────────────┼─────────────────────►
   └─loglimit───┬─0─┬───────────────────────────────────────────────┘
                └─n─┘
```

```
   ┌─lifetime──30───────┐
►──┼────────────────────┼─────────────────────────────────────────────────────────────────┤
   └─lifetime──lifetime─┘
```

PortSpecification:

```
   ┌─srcport──all──────┐ ┌─destport──all──────┐
├──┼───────────────────┼─┼────────────────────┼──────────────────────────────────────────┤
   └─srcport───┬─n───┬─┘ └─destport───┬─n───┬─┘
               ├─n─m─┤                ├─n─m─┤
               └─all─┘                └─all─┘
```

IcmpSpecification:

```
   ┌─type──all───────┐ ┌─code──all───────┐
├──┼─────────────────┼─┼─────────────────┼───────────────────────────────────────────────┤
   └─type───┬─n───┬──┘ └─code───┬─n───┬──┘
            └─all─┘             └─all─┘
```

FragmentSpecification:

```
   ┌─fragmentsonly──no─────────┐
├──┼───────────────────────────┼─────────────────────────────────────────────────────────┤
   └─fragmentsonly───┬─no──┬───┘
                     └─yes─┘
```

Defensive Filter Update Specification:

```
├──┬─────────────────────┬──┬─────────────────────┬──┬──────────────┬──┬──────────────────┬──────────────┤
   └─mode──┬─block────┬──┘  └─lifetime──lifetime─┘  └─log──┬─yes─┬─┘  └─loglimit──┬─0─┬─┘
           └─simulate─┘                                   └─no──┘                └─n─┘
```

Manual Tunnel Option:

```
              ┌─-r detail───────────┐
├──┬─display──┼─────────────────────┼──┤ Man Tunnel Sel ├──┬──────────────────┤
   │          └─-r──┬─short──┬──────┘                      │
   │                ├─detail─┤                             │
   │                └─wide───┘                             │
   ├─activate──┤ Man Tunnel Sel ├──────────────────────────┤
   └─deactivate──┬─┤ Man Tunnel Sel ├─┬────────────────────┘
                 └─-a all─────────────┘
```

Man Tunnel Selection:

```
          ┌─,────┐
          ▼      │
├──┬─ -a──Mnn────┴──────────────┬──────────────────────────┤
   │      ┌─,─────────────────┐ │
   │      ▼                   │ │
   └─ -n──IpManVpnActionName──┴─┘
```

IKE Tunnel Option:

```
                ┌─-r detail────────┐  ┌─-c current────────┐
├──┬─display────┼──────────────────┼──┼───────────────────┼──┤ IKE Tunnel Sel ├──┬──────┬──┬──────┤
   │            └─-r──┬─short──┬───┘  └─-c──┬─current─┬───┘                      └─-e───┘  │
   │                  ├─detail─┤            └─all─────┘                                    │
   │                  └─wide───┘                                                           │
   ├─deactivate──┬─┤ IKE tunnel Sel2 ├─┬───────────────────────────────────────────────────┤
   │             └─-a── all────────────┘                                                   │
   └─refresh──┤ IKE Tunnel Sel2 ├──────────────────────────────────────────────────────────┘
```

IKE Tunnel Selection:

```
          ┌─,────┐
          ▼      │
├──┬─ -a──Knn────┴──────────────────┬──────────────────────────┤
   │      ┌─,─────────────────────┐ │
   │      ▼                       │ │
   └─ -n──KeyExchangeRuleName─────┴─┘
```

IKE Tunnel Selection2:

```
        ┌─,────┐
        ▼      │
├── -a──Knn────┴───────────────────────────────────────┤
```

Dynamic Tunnel Option:

```
             ┌─-r detail──────┐  ┌─-c current─────┐
├──┬─display─┼────────────────┼──┼────────────────┼──┬────────────────────────────┬──┬──┤
   │         └─-r──┬─short──┬─┘  └─-c─┬─current─┬─┘  └─┬─-b──┤ Dyn Tunnel Sel ├─┬─┘  │
   │               ├─detail─┤         └─all─────┘      └─-s─────────────────────┘    │
   │               └─wide───┘                                                        │
   │              ┌─,───────────────────┐                                            │
   ├─activate -l──▼─LocalDynVpnRuleName─┴────────────────────────────────────────────┤
   ├─deactivate─┬─┤ Dyn Tunnel Sel2 ├─┬──────────────────────────────────────────────┤
   │            └─-a all──────────────┘                                              │
   └─refresh─┤ Dyn Tunnel Sel2 ├─────────────────────────────────────────────────────┘
```

Dyn Tunnel Selection:

```
         ┌─,───┐
├──┬─-a──▼─Ynn─┴─────────────────┬──┤
   │     ┌─,──────────────────┐  │
   ├─-n──▼─IpDynVpnActionName─┴──┤
   │     ┌─,───────────────────┐ │
   └─-l──▼─LocalDynVpnRuleName─┴─┘
```

Dyn Tunnel Selection2:

```
         ┌─,───┐
├──┬─-a──▼─Ynn─┴─────────────────┬──┤
   │     ┌─,───────────────────┐ │
   └─-l──▼─LocalDynVpnRuleName─┴─┘
```

Interface Option:

```
           ┌─-r detail──────┐
├──display─┼────────────────┼──┤
           └─-r──┬─short──┬─┘
                 ├─detail─┤
                 └─wide───┘
```

IP Traffic Test Option:

```
                                                    ┌─out─────────────────┐  ┌─-r detail──────┐
├──SrcIpAddr──DestIpAddr──┬─tcp SrcPort DestPort─┬──┼─────────────────────┼──┼────────────────┼──┤
                          ├─udp SrcPort DestPort─┤  └─┬─in SecurityClass─┬┘  └─-r──┬─short──┬─┘
                          ├─icmp─────────────────┤    └─out──────────────┘         ├─detail─┤
                          ├─icmpv6───────────────┤                                 └─wide───┘
                          ├─igmp─────────────────┤
                          ├─ipip─────────────────┤
                          ├─ah───────────────────┤
                          ├─esp──────────────────┤
                          ├─ospf─────────────────┤
                          └─n────────────────────┘
```

NATT Port Translation Option:

```
           ┌─ -r detail──────────┐
├──display─┼─────────────────────┼──┬─────────────────┬──┬───────────────────────┬──┤
           └─ -r──┬─short──┬─────┘  └─ -q─rmtIpAddr───┘  │        ┌─,─────────┐   │
                  ├─detail─┤                             │        ▼           │   │
                  └─wide───┘                             └─ -u────rmtPort─────┴───┘
```

IKED Network Security Option:

```
           ┌─ -r detail──────────┐
├──display─┼─────────────────────┼──────────────────────────────────────────────────┤
           └─ -r──┬─short──┬─────┘
                  ├─detail─┤
                  └─wide───┘
```

Network Security Server Option:

```
           ┌─ -r detail──────────┐
├──display─┼─────────────────────┼──────────────────────────────────────────────────┤
           └─ -r──┬─short──┬─────┘
                  ├─detail─┤
                  └─wide───┘
```

The z/OS UNIX ipsec command parameter descriptions

The following topics describe the individual parameter items that appear in the syntax diagram.

Rules:

- All options are case sensitive.
- Option values that are keywords are not case sensitive and can be shortened to the first three characters of the keyword, for example **-f default** can be specified as **-f DEF**.
- Option values for **-n, -g, -l**, and **-N** specify a name and are case sensitive.
- Option values for **-p** and **-z** specify a name and are not case sensitive.
- On the **ipsec -F add** command and **ipsec -F update** command, the associated values are not case sensitive. For example, **ipsec -F add srcip 10.1.1.1 dir inbound** can be specified as **ipsec -F add SRCIP 10.1.1.1 DIR INBOUND**. The associated values cannot be shortened.

The z/OS UNIX ipsec command primary options

-f
Display or modify IP filter information.

-F
Display or modify defensive filter information.

-m
Display or modify manual tunnel information.

-k
Display or modify IKE tunnel information.

-y
Display or modify dynamic tunnel information.

-i

Display interface information that is defined to the specified TCP/IP stack.

-t

Locate and display active filter rules matching particular data traffic that match the selection input.

-o Display NAT remote port translation table information.

-w Display IKE NSS client information. The Stackname options **-p** and **-z** do not apply with the -w option.

-x Display NSS server information. The Stackname option **-p** does not apply with the **-x** option.

-?

Display command help.

The z/OS UNIX ipsec command global options

-d

Generates debug information during command execution. You can specify an optional *debuglevel* value when you use the **-d** option. Debug output is sent to stderr or to stdout, determined by the debug level. Debug information accumulates with the higher levels (for example debug level 3 also includes the information from level 1 and level 2).

debuglevel

The following debug levels are available:

1 Generate functional level debug information in stdout in a formatted form.

The functional level debug can be specified with any option. If there is functional level debug information for that report, it is displayed in addition to the base report (such as **-o** report). If not, only the base report is displayed.

2 Generate general debug information in stderr in an unformatted form for criteria that was specified when the command was issued.

You can specify the selective level debug with any option. Selective debug information is available only for **-f**, **-F, -m, -y** (without **-b**), and **-o** reports. For other reports, only the base report is displayed.

3 Generate operational level debug information in stderr in an unformatted form. This is the default debug level.

The z/OS UNIX ipsec command stackname options

-p *stackname*

Selects a local stack. The *stackname* parameter specifies the name of the TCP/IP address space. If the **-p** option is not specified, the default stack is selected. The default stack refers to the default TCP/IP address space that is specified on the TCPIPJOBNAME statement in the resolver configuration data set.

-z *nsclientname*

Selects a NSS client. The *nsclientname* parameter specifies the name of a NSS client as specified on the ClientName parameter for the NssStackConfig statement in the IKED.CONF file on the client system. If not specified on the NssStackConfig statement, it defaults to the form of *systemname_stackname*. To produce a list of available NSS clients, issue the **ipsec -x** command. There is no default for this parameter.

The z/OS UNIX ipsec command target options

-p *stackname*

Indicates that the defensive filter or filters are stack-specific filters that are associated with the local TCP/IP stack specified by the *stackname* value. The *stackname* parameter specifies the name of the TCP/IP address space. If neither the **-p** nor the **-G** is specified, the default stack is selected. The default stack is the default TCP/IP address space that is specified on the TCPIPJOBNAME statement in the resolver configuration data set.

Results: To successfully add a stack-specific defensive filter, the following conditions must be met:

- There must be a DmStackConfig statement for stack *stackname* in the DMD configuration file with a mode of Active or Simulate.
- The stack must support IP security.

-G Indicates that the defensive filter or filters are global filters that apply to all TCP/IP stacks that are listed in the DMD configuration file and that support IP security on the local system.

Results:

- When you add a global defensive filter, the DMD maintains a copy of the global filter. A stack-specific copy is generated from the global filter and installed in each local TCP/IP stack that is listed in the DMD configuration file and that supports IP security.
- When you display global defensive filters, the global copy of the defensive filters is displayed. The global copy of the filter does not contain any accumulated counts that are kept by each TCP/IP stack. Issue the command with the **-p** *stackname* option to display the accumulated counts for a specific stack.
- When you update a global defensive filter, the update is applied to the global filter and to each of the generated stack-specific copies.
- When you delete global defensive filters, the global filter or filters and each of the stack-specific copies that are generated are deleted.
- When a stack-specific copy of a global filter is updated with **-p** (stack specific) option, only that copy of the filter is updated. If you make a subsequent update to the global filter with the **-G** option, all copies of the filter are updated, including the one that was previously updated with the **-p** *stackname* option. The last update always remains in effect.
- When you delete a stack-specific copy of a global filter or it expires, that copy is no longer affected by updates to the global filter.
- When a global filter expires before one or more of its stack-specific copies expires, you can still perform global update and delete operations. The expired global filter is retained to allow the global update and delete operations. Stack-specific copies of the expired global filter are not installed in new stacks that start up. The expired global copy is removed completely when all stack-specific copies expire or are deleted.
- An expired global filter is displayed with the State value `Pending Inactive` and the LifetimeExpires value `Expired` while one or more of its stack-specific copies is still active.

The z/OS UNIX ipsec command IP filter (-f) option

z/OS UNIX ipsec command IP filter (-f) option parameters:

display

Displays the selected IP filters. If no filters are selected, then all filter rules (with respect to the display scope) are displayed.

-r *format*

Displays IP filter information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

-c *scope*

Displays the scope. The default scope is `current`.

current

Display filter rules that are current and that are in use by the stack. IP security filter rules that are inactive because of time conditions are not included. If any defensive filter rules are in use by the stack, they are included.

policy

Displays filter rules that are configured from the policy definition. Filter rules that are inactive because of time conditions are not included. The filter rules that are displayed using this option might or might not be current at the stack. This option also displays global policy settings. Defensive filter rules are not included.

profile

Displays filter rules that are configured as default filter rules from the IPSEC statement on the TCPIP profile. The filter rules that are displayed using this option might or might not be current at the stack. Defensive filter rules are not included.

-a M*nn* **or -a Y***nn*

Displays the IP security filters that are associated with the specified tunnel IDs. The tunnel IDs must have an M (manual) prefix or a Y (dynamic) prefix.

For manual tunnels, multiple filter rules might be associated with a manual tunnel ID. There is a one-to-one correspondence between a manual tunnel ID and an IpManVpnAction definition. If multiple active filter rules reference an IpManVpnAction, they are all displayed. Filter rules that are inactive because of time conditions are not included.

Tip: To display all of the statically defined dynamic anchor filters, specify **-a Y0**.

-h For any displayed NAT Traversal (NATT) anchor filter, the associated NAT resolution filters (NRFs) are also displayed.

-n *IpFilterRuleName*

Specifies one or more IP security filters to be selected. The names used must correspond to either IpFilterRule names that are specified in a Policy Agent configuration file or to the stack-generated names assigned to the default rules in the TCP/IP profile. The IpFilterRule base name might refer to more than one filter rule in the selected stack. In this case, the base name has an appended number that uniquely identifies the generated rules. These names have the following format:

name:*index*

name

The base name.

index
An integer that is assigned with the filter rule. The integer corresponds to the order in which the filter rule was generated from its base IpFilterRule statement.

For the command `ipsec -f display -n` *IpFilterRuleName*, all IpFilterRule statements (with respect to the display scope) with a base name that matches the *IpFilterRuleName* value are displayed.

-N *DefensiveFilterName*
Specifies one or more defensive filters to be selected. The names used must correspond to defensive filter rule names that are specified when the defensive filters are added.

Tip: The DefensiveFilter base name might refer to more than one filter rule in a stack. In this case, the base name has an appended number that uniquely identifies the generated defensive filter. These names have the following format:

name
The base name.

index
An integer that is assigned to the filter rule.

For the command `ipsec -f display -N` *DefensiveFilterName*, all defensive filters (with respect to the display scope) with a base name that matches the *DefensiveFilterName* value are displayed.

-g *IpFilterGroupName*
Specifies one or more IP security filter groups that are to be displayed. The names that are specified must correspond to the IpFilterGroup names that are specified in a Policy Agent configuration file.

default
Causes the stack to use the default IP security filter rules. Default IP filter rules consist of the IP filter rules that are specified by the TCPIP profile, if any, and an implicit DENY-ALL filter rule. If other IP filters were in use as generated from a policy configuration file (policy IP filters), those IP filters remain intact, but are not used by the stack. While the profile IP filters are in effect, manual, dynamic, and IKE tunnels still exist, but they are not used. These tunnels might expire or be deactivated, but cannot cleanly terminate with the peer. Tunnel refreshes might not occur, and new dynamic tunnels might not be activated. If present, defensive filters remain in use by the stack along with the default IP security filter rules.

Note: The request to switch to the default profile IP filters is remembered across activations of the stack and system IPLs.

reload
Causes the stack to use the policy IP security filter rules that are supplied from a policy configuration file. If no policy IP filters were previously defined to the stack, the stack continues to use the default IP filter rules until the policy configuration file is installed by the Policy Agent. If policy IP filter rules were previously defined to the stack, those policy IP filters become effective again. Tunnel activity resumes, including refreshes and new activations. If the IKE daemon is active, it attempts to perform all automatic activations that are configured. If present, defensive filters remain in use by the stack along with the policy IP security filter rules.

Note: The request to switch to the policy IP filter rules is remembered across activations of the stack and system IPLs.

Tip: Displays of IP filter rules indicate whether the IP security rules originate as default profile rules or as policy rules.

See also "IP filter (-f) primary option" on page 738 for report details and examples.

The z/OS UNIX ipsec command defensive filter (-F) option

The following parameters can be used with the z/OS UNIX **ipsec** command defensive filter (**-F**) option.

z/OS UNIX ipsec command defensive filter (-F) option parameters:

`display`
: Displays the selected defensive filters. If no filters are selected, then all defensive filter rules are displayed.

`-r format`
: Displays defensive filter information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

`-N` *`DefensiveFilterName`*
: Specifies one or more defensive filters to be selected. The names used must correspond to defensive filter rule names that are specified when the defensive filters are added.

Tip: The DefensiveFilter base name can refer to more than one filter rule in a stack. In this case, the base name has an appended number that uniquely identifies the defensive filter that is generated. These names have the following format:

`name`
: The base name.

`index`
: An integer that is assigned to the filter rule.

The command `ipsec -F display -N` *`DefensiveFilterName`* displays all defensive filters with a base name that matches the *DefensiveFilterName* value.

`add`
: Adds a defensive filter to the top of the defensive filters search list. You cannot add an IP security filter with this option; it must be configured in the TCPIP profile or in a policy configuration file. The following add parameters determine the characteristics of the added defensive filter:

`srcip`
: A source IP address specification. Possible values are:

`ipaddress`
: A single IP address. This value indicates the source address that must be contained in an IP packet for the packet to match this filter rule.

`ipaddress/prefixLength`
: A prefix address specification that indicates the applicable source IP addresses that can be contained in an IP packet for the packet to match this filter rule. The *prefixLength* value is the number of unmasked leading bits in the *ipaddress* value. The *prefixLength* value can be in the

range 0-32 for IPv4 addresses and in the range 0-128 for IPv6 addresses. An IP packet matches this condition if the unmasked bits of its source address are identical to the defined unmasked bits.

`all`
Indicates that the filter rule applies to any source IP address. This is the default value.

Rule: If both the srcip and destip parameters are specified, the IP addresses must be in the same family (IPv4 or IPv6).

`destip`
A destination IP address specification. Possible values are:

`ipaddress`
A single IP address. This value indicates the destination address that must be contained in an IP packet for the packet to match this filter rule.

`ipaddress/prefixLength`
A prefix address specification that indicates the applicable destination IP addresses that can be contained in an IP packet for the packet to match this filter rule. The *prefixLength* value is the number of unmasked leading bits in the *ipaddress* value. The *prefixLength* value can be in the range 0-32 for IPv4 addresses and in the range 0-128 for IPv6 addresses. An IP packet matches this condition if the unmasked bits of its destination address are identical to the defined unmasked bits.

`all`
Indicates that the filter rule applies to any destination IP address. This is the default value.

Rule: If both the srcip and destip parameters are specified, the IP addresses must be in the same family (IPv4 or IPv6).

`prot`
The IP protocol that must be contained in an IP packet for the packet to match this filter rule. If an *n* value is specified, it identifies a protocol number. The value for *n* can be in the range 0-255. If the value *all* is specified, then the filter rule applies to any protocol. The default value is `all`.

The protocol specification Opaque matches any IPv6 packet for which the upper-layer protocol is not known because of fragmentation. This specification always matches non-initial fragments, and it also matches initial fragments if the upper-layer protocol value is not included in the first fragment. Use of the Opaque protocol specification is applicable only to routed fragments because, for all local traffic, the stack applies IP filter rules only to fully assembled packets.

Rule: The protocol specification Opaque can be used only for IPv6 addresses.

`srcport`
If the protocol TCP or UDP is specified, then you can specify a srcport value. The srcport value indicates the source ports that must be contained in an IP packet for the packet to match this filter rule.

Valid values for *n* are in the range 1-65535. If an *m* value is specified, it must be greater than or equal to the *n* value and less than 65536. If the value all is specified, then the filter rule applies to any source port. The default value is all.

Restriction: If the Routing parameter value is Routed or Either, you must use either the default srcport value or the value `all`.

`destport`

If the protocol TCP or UDP is specified, then you can specify a destport value. The destport value indicates the destination ports that must be contained in an IP packet for the packet to match this filter rule.

Valid values for *n* are in the range 1-65535. If an *m* value is specified, it must be greater than or equal to the *n* value and less than 65536. If the value `all` is specified, then the filter rule applies to any destination port. The default value is `all`.

Restriction: If the Routing parameter value is Routed or Either, you must use either the default destport value or the value `all`.

`type`

If the protocol ICMP or ICMPv6 is specified, then you can specify a type value. The type value indicates the ICMP type that must be contained in an IPv4 ICMP packet or an IPv6 ICMPv6 packet for the packet to match this filter rule. Valid values for *n* are in the range 0-255. If the value `all` is specified, then the filter rule applies to any ICMP type. The default value is `all`.

Restriction: If the Routing parameter value is Routed or Either, you must use either the default type value or the value `all`.

`code`

If the protocol ICMP or ICMPv6 is specified, then you can specify a code value. The code value indicates the ICMP code that must be contained in an IPv4 ICMP packet or an IPv6 ICMPv6 packet for the packet to match this filter rule. Valid values for *n* are in the range 0-255. If you specify the value `all`, then the filter rule applies to any ICMP code. The default value is `all`.

Restriction: If the Routing parameter value is Routed or Either, you must use either the default code value or the value `all`.

`dir`

The direction a packet must take for the packet to match this filter rule. Valid values are:

`inbound`

Indicates that this filter rule applies to inbound packets. This is the default.

`outbound`

Indicates that this filter rule applies to outbound packets.

`routing`

The routing characteristics that a packet must have for the packet to match this filter rule. Valid values are:

`local`

Indicates that this filter rule applies to packets that are destined for this stack or that originate from this stack. This is the default.

routed
Indicates that this filter rule applies to packets that are being forwarded by this stack.

either
Indicates that this filter rule applies to forwarded and non-forwarded packets.

fragmentsonly
When set to Yes, this filter rule matches only fragmented packets. When set to No, this filter rule matches both fragmented packets and non-fragmented packets. Fragments are matched only in routed traffic, because the TCP/IP stack applies IP filter rules for local traffic only to fully reassembled packets.

Tip: Use this keyword to block all fragmented traffic.

mode
The defensive filter mode. The default value is block.

block
Indicates that the defensive filter blocks or denies packets that match the characteristics of the filter.

simulate
Indicates that the defensive filter simulates a block. If a packet matches a defensive filter with the mode value simulate, a log record is written to syslog indicating that the packet would have been denied by this filter. The packet is not denied and IP filtering continues.

Rule: If the mode value `Simulate` is configured for a TCP/IP stack in the DMD configuration file, that value overrides the individual defensive filter mode setting. For example, if a defensive filter with the mode value `block` is added to a stack and the DmStackConfig statement for that stack has a configured mode of `Simulate`, a packet that matches the defensive filter is not blocked. Instead, a block is simulated. The defensive filter retains the block mode. If the mode value in a DmStackConfig statement for the stack is updated to `Active`, a packet that matches the defensive filter is blocked.

log
The logging action for a defensive filter.

yes
A log record is written when a packet matches this filter rule. This is the default.

no A log record is not written when a packet matches this filter rule.

Restriction: If the mode parameter value is `simulate`, the log parameter must be set to the value yes.

loglimit
The log limit for a defensive filter. The loglimit value is used to enable or disable the limiting of defensive filter match messages (EZD1721I and EZD1722I) written to syslogd. For more information, see filter-match logging in z/OS Communications Server: IP Configuration Guide.

0 Disables the limiting of defensive filter match messages written to syslogd. If logging is being done for this defensive filter, a message is generated for each packet that matches the defensive filter.

n Enables the limiting of defensive filter-match messages written to syslogd. Valid values are in the range 1 - 9999. The value specifies the limit of the average rate of filter-match messages generated in a 5-minute interval for a defensive filter. For example, a value of 100 limits the average rate of filter-match messages to 100 messages per 5-minute interval. A burst of up to 100 messages is allowed while maintaining the long-term average of 100 messages per 5-minute interval.

Result: If **`loglimit`** is not specified, the default value is the DefaultLogLimit value specified in the DMD configuration file. See z/OS Communications Server: IP Configuration Reference for more information about the DefaultLogLimit keyword.

`lifetime`

The length of time, in minutes, that the defensive filter remains in use. Valid values are in the range 1-20160. The default value is 30 minutes.

Tip: If the lifetime value exceeds the maximum lifetime value that is configured for a stack, the defensive filter's lifetime value is set to the maximum allowed lifetime value. The maximum lifetime value is configured with the MaxLifetime keyword in the DMD configuration file. See z/OS Communications Server: IP Configuration Reference for more information about the MaxLifetime keyword.

Results:

- If you specified the value `all` (or is in effect by default) for both the srcip and destip parameters, a defensive filter is added to match any IPv4 source and destination address. If a stack supports IP security for IPv6, a defensive filter is also added to match any IPv6 source and destination address. If both an IPv4 and IPv6 filter are installed, the base name is the name that was specified when the filter was added. Different index values are assigned to each filter rule by the DMD.
- If you specified the value `all` for either the scrip or destip parameter and a specific address family is specified for the other parameter, a defensive filter is added for the specific address family. For example, srcip all and destip 10.1.1.1 result in a defensive filter being added to match any IPv4 source address and a destination address of 10.1.1.1.
- If both an IPv4 and IPv6 filter are installed and the protocol value is icmp or 1, type and code values are relevant only for the IPv4 filter. The IPv6 filter does not use the type and code values to determine whether an IPv6 packet matches the filter.
- If both an IPv4 and IPv6 filter are installed and the protocol value is icmpv6 or 58, type and code values are relevant only in the IPv6 filter. The IPv4 filter does not use the type and code values to determine whether an IPv4 packet matches the filter.
- If both an IPv4 and IPv6 filter are installed, the loglimit is applied both to the IPv4 filter and to the IPv6 filter.
- If a defensive filter add specifies IPv6 addresses, the filter is added only to a stack that supports IP security for IPv6.

`-N` *`DefensiveFilterName`*

A string 1-32 characters in length that specifies the name of the defensive filter that is being added. The name cannot start with a dash (-). The name also cannot contain any commas (,). A comma is treated as delimiter by the **`ipsec`** command and it is therefore ignored.

Tip: Global and stack-specific defensive filters share the same name space; therefore, a filter name cannot be used for both a global filter and a stack-specific filter. If you are manually creating defensive filters, avoid conflicts between global and stack-specific filter names by selecting a distinct naming convention for each. For example, start all global filter names with the letter G.

update
Updates a defensive filter's characteristics. You cannot update an IP security filter with this option. You must update the IP security filter in the TCPIP profile or in a policy configuration file. You can modify the following defensive filter characteristics:

mode
The defensive filter mode. Valid values are:

block
Indicates that the defensive filter blocks or denies packets that match the characteristics of the filter.

simulate
Indicates that the defensive filter simulates a block. If a packet matches a defensive filter with the mode `simulate`, a log record is written to syslog indicating that the packet would have been denied by this filter. The packet is not denied and IP filtering continues.

Rule: If the mode value Simulate is configured for a TCP/IP stack in the DMD configuration file, it overrides the individual defensive filter's mode setting. For example, if a defensive filter is updated to be in block mode and the DmStackConfig statement for the stack where the filter is installed has a configured mode of Simulate, a packet that matches the defensive filter is not blocked. Instead, a block is simulated. The defensive filter retains the block mode. If the DmStackConfig statement for the stack is updated to be in Active mode, a packet that matches the defensive filter is blocked.

log
The logging action for a defensive filter.

yes
A log record is written when a packet matches this filter rule.

no A log record is not written when a packet matches this filter rule.

Restrictions:

- If the mode value is simulate and the log parameter is specified, the log value must be configured as yes.
- If the mode parameter is not specified and the filter's mode is simulate, the log value (if it is specified) must be configured as yes.

Result: If mode is simulate and log is not specified, the log value is set to yes in the filter.

loglimit
The log limit for a defensive filter. The loglimit value is used to enable or disable the limiting of defensive filter match messages (EZD1721I and EZD1722I) written to syslogd. For more information, see filter-match logging in z/OS Communications Server: IP Configuration Guide.

0 Disables the limiting of defensive filter match messages written to syslogd. If logging is being done for this defensive filter, a message is generated for each packet that matches the defensive filter.

n Enables the limiting of defensive filter-match messages written to syslogd. Valid values are in the range 1 - 9999. The value specifies the limit of the average rate of filter-match messages generated in a 5-minute interval for a defensive filter. For example, a value of 100 limits the average rate of filter-match messages to 100 messages per 5-minute interval. A burst of up to 100 messages is allowed while maintaining the long-term average of 100 messages per 5-minute interval.

lifetime
The additional time (in minutes) that the defensive filter remains in use from the time the update command is processed.

-N *DefensiveFilterName*
Specifies the name of the defensive filter that is to be updated. The name must correspond to the defensive filter rule name that was specified when the defensive filter was added.

delete
Deletes one or more defensive filters. You cannot delete an IP security filter with this option. You must remove IP security filters from the TCPIP profile or the policy configuration file.

-N *DefensiveFilterName*
Specifies one or more defensive filters that are to be deleted. The names must correspond to defensive filter rule names that were specified when the defensive filters were added.

-N all
Specifies that all defensive filters are deleted on the target stack (**-p**) or deleted globally (**-G**).

The z/OS UNIX ipsec command manual tunnel (-m) option

z/OS UNIX ipsec command manual tunnel (-m) option parameters:

display
Displays the selected manual tunnels. Manual tunnels that are inactive because of time conditions are not available for display by **ipsec**. The **pasearch** command can be used to display configured manual tunnels with their time conditions.

All manual tunnels that are installed in the stack are displayed. For a tunnel to be in use protecting IP traffic, the following must be true:

- The current filter set must be the policy set.
- An active filter must reference the tunnel (IpManVpnAction).
- The tunnel state must be active.

To determine whether the manual tunnel is in use, issue the `ipsec -f display -a Mxx` command, where **M***xx* is the manual tunnel ID from the tunnel display.

-r *format*
Displays IP security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

activate
Activates the selected manual tunnels. If no manual tunnels are selected, then

all manual tunnels are activated. IP traffic is protected by the algorithms that are defined by the manual tunnels. If the default filter set (defined in the TCPIP stack profile) is active, then the tunnel state cannot be changed and the activate is rejected.

deactivate
Deactivates the selected manual tunnels. The result is that IP traffic that would have used the manual tunnel is discarded while the manual tunnel is inactive.

-a M*nn* or -a all
Selects one or more manual tunnels that are associated with the specified tunnel ID. The tunnel IDs must have an M (manual) prefix.

-a all option is valid only with the **deactivate** parameter and indicates that all manual tunnels are to be deactivated.

-n *IpManVpnActionName*
Specifies one or more manual tunnels to be selected. The names that are used must correspond to IpManVpnAction names that are specified in a Policy Agent configuration file.

See also "Manual tunnel (-m) primary option" on page 761 for report details and examples.

The z/OS UNIX ipsec command IKE tunnel (-k) option

z/OS UNIX ipsec command IKE tunnel (-k) option parameters:

display
Displays security association (SA) data associated with the selected IKE tunnels.

-r *format*
Displays IP security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

-c *scope*
Selects the scope of information displayed. The default scope is `current`.

current
Displays IKE tunnel information about current IKE SAs only. When the selection criteria specifies a name (KeyExchangeRuleName), multiple current SAs can correspond to the specified name.

all
Displays IKE tunnel information about SAs, including SAs that might no longer be in use. This includes SAs that have expired and have not been garbage collected. It also includes SAs that have not yet expired but that have been superseded by a refresh.

-e Use this (cascade) option to additionally display the dynamic SAs that are associated with the specified IKE SAs. When the cascade option is used, dynamic SA information is obtained from the IKE server (and not from the stack). The display scope does not apply to the dynamic SAs that are reported as a result of the cascade option.

deactivate
Deactivates the selected IKE tunnels. The IKE tunnel is terminated for subsequent negotiations. To indicate all IKE tunnels, specify **-a all** on the command. New IKE SAs can be established as needed (for example, to support on demand or command requests).

Note:

1. All dynamic tunnels that are associated with deactivated IKE tunnels are also deactivated as part of this request.
2. If **-a K0** is specified with **-k deactivate**, all IKE tunnels with the tunnel ID K0 (which indicates an IKE tunnel with a current state not equal to Active) are deactivated.

Restriction: Use this option only if there is concern that the cryptographic keys in use on a current SA have been compromised. Reactivating IKE tunnels is a processor-intensive operation. If the scope of a deactivate request is large, then overall system performance could be affected.

`refresh`
Refreshes the cryptographic keys for the selected IKE tunnels by performing a reauthentication operation to negotiate a new IKE tunnel. Configuration options typically cause a refresh (and therefore a new set of cryptographic keys) on a lifetime or lifesize basis.

Restriction: Do not use this refresh option unless the IKE tunnel appears to be in a state that keeps it from being used. Refreshing IKE tunnels is a processor-intensive operation. If the scope of a refresh request is large, then overall system performance could be affected.

`-a K`*`nn`* `or -a all`
Selects one or more IKE tunnel IDs. The tunnel IDs must have a K (IKE) prefix. The **-a all** option is valid only with the **deactivate** parameter and indicates that all IKE tunnels are to be deactivated.

`-n` *`KeyExchangeRuleName`*
Specifies one or more IKE tunnels to be selected. The names that are specified must correspond to KeyExchangeRule statements that are specified in a Policy Agent configuration file.

See also “IKE tunnel (-k) primary option” on page 764 for report details and examples.

The z/OS UNIX ipsec command dynamic tunnel (-y) option

z/OS UNIX ipsec command dynamic tunnel (-y) option parameters:

`display`
Displays the security association (SA) data that is associated with the selected dynamic tunnels. The display reflects information from the SA and is not about specific systems or resources that are being protected by the SA. More information about the resources being protected can be determined by displaying the filter rules in place that correspond to a specific dynamic tunnel (**ipsec -f display -a Y***nn*). Unless the **-b** option is specified, the dynamic SA information is obtained from the stack. Shadow tunnels are displayed only when the **-s** option is specified.

`-r` *`format`*
Display IP Security information in a given format. The default format is `detail`. See “The ipsec command general report concepts” on page 723 for a description of the different report formats.

`-c` *`scope`*
Displays the scope. The default scope is `current`.

current
Displays dynamic SA information about current SAs only. When selection criteria specifies a name (LocalDynVpnRuleName or IpDynVpnActionName), multiple current SAs can correspond to the specified name.

all
Displays dynamic SA information about SAs, including SAs that might no longer be in use. This includes SAs that have been superseded by a refresh.

-b Dynamic tunnel information for display comes from the specified stack, unless this option is also specified. With this option, the dynamic tunnel information for display comes from the IKE daemon.

-n *IPDynVpnActionName*
Specifies one or more dynamic tunnels that are to be selected. The names that are specified must correspond to IpDynVpnAction names that are specified in a Policy Agent configuration file.

-s Displays shadow dynamic tunnel SAs from the stack. Shadow security associations are used by the sysplex-wide security associations (SWSA) function to distribute security associations to target stacks of distributed DVIPAs. See the considerations for sysplex-wide security associations information in the z/OS Communications Server: IP Configuration Guide for details.

activate
Activates one or more dynamic tunnels identified in a LocalDynVpnRule that is defined by a Policy Agent.

Rule: On the **activate** option you cannot specify an IpDynVpnAction name or tunnel ID.

deactivate
Deactivates the selected dynamic tunnels. To indicate all dynamic tunnels, specify **-a all** on the command. The dynamic tunnel becomes unavailable for IP traffic. New dynamic SAs can be established as needed (for example, on-demand or command requests).

Note: If **-a Y0** is specified with **-y deactivate**, all dynamic tunnels with the tunnel ID Y0 (which indicates a dynamic tunnel with a current state not equal to DONE) are deactivated.

Restriction: Use this option only if there is concern that the cryptographic keys that are in use on the current SA have been compromised. Reactivating dynamic tunnels is a processor-intensive operation. If the scope of a deactivate request is large, then overall system performance can be affected.

refresh
Refreshes the cryptographic keys for the selected dynamic tunnels. Configuration options typically cause a refresh (and therefore a new set of cryptographic keys) on a lifetime or lifesize basis.

Restriction: Do not use this refresh option unless the current dynamic SA appears to be in a state that keeps it from being used for IP traffic. Refreshing dynamic tunnels is a processor intensive operation. If the scope of a refresh request is large, then overall system performance can be affected.

-a Y*nn* or -a all
Selects one or more dynamic tunnel IDs. The tunnel IDs must have a Y (dynamic) prefix.

Rule: The **-a all** option is valid only with the **deactivate** parameter and indicates that all dynamic tunnels are to be deactivated.

-l *LocalDynVpnRuleName*
Specifies one or more startable resource specifications for which dynamic SAs are to be established. The names that are specified must correspond to the LocalDynVpnRule names that are specified in a Policy Agent configuration file. A LocalDynVpnRule name becomes associated with a dynamic tunnel only through an **ipsec** command activation request or through autoactivation. Only those dynamic tunnels can be referenced with a LocalDynVpnRule name. Dynamic tunnels that were started by other means (for example, on-demand activation or as responder) have no LocalDynVpnRule name association and cannot be referenced with the **-l** option.

See also "Dynamic tunnel (-y) primary option" on page 772 for report details and examples.

The z/OS UNIX ipsec command interface (-i) option

z/OS UNIX ipsec command interface (-i) option parameters:

display
Displays interface information that is defined to the specified TCP/IP stack.

-r *format*
Displays IP security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

See also "Interface (-i) primary option" on page 789 for report details and examples.

The z/OS UNIX ipsec command IP traffic test (-t) option

z/OS UNIX ipsec command IP traffic test (-t) option parameters:

SrcIpAddr
The source IP address of the traffic to be tested or protected.

DestIpAddr
The destination IP address of the traffic to be tested or protected.

Protocol Specification
A protocol keyword can be selected from those shown in the syntax diagram, or a protocol number of the traffic to be tested. The IP traffic test matches on protocol when the IP filter contains the same protocol number or when the IP filter applies to all protocols.

SrcPortDestPort
If the TCP or UDP protocol keywords are specified, then source and destination port numbers must be supplied. Port number 0 indicates to match any port.

For traffic that traverses a NAT, an internal remote port translation function is used in some cases to increase usability. Remote port translation is applicable only to ephemeral ports (ports in the range 1024 - 65 535). If the remote port translation function is being used, then there is both an original remote port value and a translated remote port value. The traffic test treats the input remote port (source port for an inbound packet, destination port for an outbound packet) as the original port value. In most cases when remote port translation is performed, the specific port value is not known and the value 0 should be specified on input to the traffic test. For more details about NAT traversal and remote port translation, see the remote port translation information in the z/OS Communications Server: IP Configuration Guide.

Direction Specification

The traffic direction can be specified as in or out. If the traffic direction keyword is not specified, then both in and out directions are used.

SecurityClass

If the traffic direction keyword *in* is specified, then a security class must be supplied. A SecurityClass value of 0 indicates to match any security class.

-r *format*

Displays IP Security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

See also "IP traffic test (-t) primary option" on page 791 for report details and examples.

The z/OS UNIX ipsec command NATT port translation (-o) option

z/OS UNIX ipsec command NATT port translation (-o) option parameters:

display

Display the selected NAT traversal remote port translations. If no selected remote IP addresses are specified, all of the NAT traversal remote port translations are displayed by default. If there is a selected remote IP address (using the **-q** option), or if there is a selected remote IP address with one or more ports (using the **-q -u** options), then the selected NAT Traversal remote port translation information is displayed.

-r

Displays IP Security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" on page 723 for a description of the different report formats.

-q *rmtIpAddr*

Displays the NAT traversal remote port translation information associated with the given remote IP addresses.

-u *rmtPort*

Displays the NAT traversal remote port translation information associated with the given remote ports. Valid values for *rmtPort* are in the range 1 - 65 535. The specified port value is matched against both the original port value and the translated port value.

See also "NATT port translation (-o) primary option" on page 801 for report details and examples.

The z/OS UNIX ipsec command IKED network security (-w) option

z/OS UNIX ipsec command IKED network security (-w) option parameters:

`display`
Display network security configuration information for the active stacks on the local system.

`-r` *`format`*
Display network security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" for a description of the different report formats.

The z/OS UNIX ipsec command network security server (-x) option

z/OS UNIX ipsec command network security server (-x) option parameters:

`display`
Display information for each NSS IPSec client that is currently connected to the NSS server. When the **-z** option is specified, only information for the requested client is returned; otherwise, information is returned for each IPSec client that is connected to the server.

`-r` *`format`*
Display network security information in a given format. The default format is `detail`. See "The ipsec command general report concepts" for a description of the different report formats.

The ipsec command report details and examples

This material contains descriptive information about the formatting and contents of **ipsec** reports, including examples.

The ipsec command general report concepts

In order to fully understand the following concepts and fields, you need some general knowledge of IP security. See IP security in z/OS Communications Server: IP Configuration Guide for more information about IP security. Also, see defensive filtering information in z/OS Communications Server: IP Configuration Guide for details about defensive filters.

The ipsec command report format:
The **-r** option controls the output format of any display report: short, detail (default), and wide. The reported data is the same for all three report formats with the difference being the layout of the field headings and field values.

Tip: When the **-z** option is specified, the stack name on the first line of the report is changed from `Stack Name` to `NSS Client Name`.

`short`
Displays IP security information in short summary format. Short format displays minimal information on the screen in a vertical orientation. Each entry can span multiple lines. The heading lines for the record type are displayed once (and first), and contain a descriptive label for each record field that is displayed. Following the heading line, each record is displayed with one or more fields per line, arranged so that the primary name associated with the entry appears first and positionally separates the entries. Both the heading lines and the entry lines use a vertical bar (`|`) as a field separation character that delimits each value. The following example shows a short format.

```
ipsec -p tcpcs4 -f display -r short

CS V2R1 ipsec  Stack Name: TCPCS4  Tue Feb 14 06:53:45 2012
Primary:  Filter          Function: Display            Format:   Short
Source:   Stack Policy    Scope:    Current            TotAvail: 164
Logging:  On              Predecap: Off                DVIPSec:  Yes
NatKeepAlive:  20         FIPS140:  No
Defensive Mode: Inactive

FilterName  |FilterNameExtension
            |GroupName
            |LocalStartActionName
            |VpnActionName
            |TunnelID
            |Type|DefensiveType|State|Action|Scope|Direction|OnDemand
            |SecurityClass|Logging|LogLimit
            |Protocol|ICMPType|ICMPTypeGran|ICMPCode|ICMPCodeGran
            |OSPFType|TCPQualifier|ProtocolGran
            |SourceAddress
            |SourceAddressPrefix
            |SourceAddressRange
            |SourceAddressGran
            |SourcePort|SourcePortRange|SourcePortGran
            |DestAddress
            |DestAddressPrefix
            |DestAddressRange
            |DestAddressGran
            |DestPort|DestPortRange|DestPortGran
            |OrigRmtConnPort|RmtIDpayload|RmtUdpEncapPort
            |CreateTime|UpdateTime
            |DiscardAction|MIPv6Type|MIPv6TypeGran|TypeRange|CodeRange
            |RemoteIdentityType|RemoteIdentity
            |FragmentsOnly|FilterMatches|LifetimeExpires|AssociatedStackCount
IPSecGWv4~7 |1
            |n/a
            |IPSecGWv4~6
            |ESP-Hmac_Md5-AES
            |Y6
            |Dynamic|n/a|Active|Permit|Routed|Outbound|Yes
            |0|All|n/a
            |TCP(6)|n/a|n/a|n/a|n/a
            |n/a|None|Rule
            |10.81.2.0
            |24
            |n/a
            |Packet
            |All|n/a|Rule
            |10.81.8.1
            |n/a
            |10.81.8.6
            |Packet
            |All|n/a|Rule
            |n/a|n/a|n/a
            |n/a|n/a
            |Silent|n/a|n/a|n/a|n/a
            |n/a|n/a
            |No|0|n/a|n/a
IPSecGWv4~7 |1
            |n/a
            |IPSecGWv4~6
            |ESP-Hmac_Md5-AES
            |Y0
            |Dynamic Anchor|n/a|Active|Permit|Routed|Outbound|Yes
            |0|All|n/a
            |TCP(6)|n/a|n/a|n/a|n/a
            |n/a|None|Rule
            |10.81.2.0
```

```
             24
             n/a
             Packet
             All|n/a|Rule
             10.81.8.1
             n/a
             10.81.8.6
             Packet
             All|n/a|Rule
             n/a|n/a|n/a
             2008/02/11 18:13:12|2008/02/11 18:13:12
             Silent|n/a|n/a|n/a|n/a
             n/a|n/a
             No|0|n/a|n/a
IPSecGWv4~7  2
             n/a
             IPSecGWv4~6
             ESP-Hmac_Md5-AES
             Y6
             Dynamic|n/a|Active|Permit|Routed|Inbound|Yes
             0|All|n/a
             TCP(6)|n/a|n/a|n/a|n/a
             n/a|None|Rule
             10.81.8.1
             n/a
             10.81.8.6
             Packet
             All|n/a|Rule
             10.81.2.0
             24
             n/a
             Packet
             All|n/a|Rule
             n/a|n/a|n/a
             n/a|n/a
             Silent|n/a|n/a|n/a|n/a
             n/a|n/a
             No|0|n/a|n/a
IPSecGWv4~7  2
             n/a
             IPSecGWv4~6
             ESP-Hmac_Md5-AES
             Y0
             Dynamic Anchor|n/a|Active|Permit|Routed|Inbound|Yes
             0|All|n/a
             TCP(6)|n/a|n/a|n/a|n/a
             n/a|None|Rule
             10.81.8.1
             n/a
             10.81.8.6
             Packet
             All|n/a|Rule
             10.81.2.0
             24
             n/a
             Packet
             All|n/a|Rule
             n/a|n/a|n/a
             2008/02/11 18:13:12|2008/02/11 18:13:12
             Silent|n/a|n/a|n/a|n/a
             n/a|n/a
             No|0|n/a|n/a
IPSecGWv6~7  1
             n/a
             IPSecGWv6~6
             ESP-Hmac_Md5-AES
             Y3
```

```
              Dynamic|n/a|Active|Permit|Routed|Outbound|Yes
              0|All|n/a
              TCP(6)|n/a|n/a|n/a|n/a
              n/a|None|Rule
              2001:db8:10::81:2:0
              112
              n/a
              Packet
              All|n/a|Rule
              2001:db8:10::81:8:1
              n/a
              2001:db8:10::81:8:6
              Packet
              All|n/a|Rule
              n/a|n/a|n/a
              n/a|n/a
              Silent|n/a|n/a|n/a|n/a
              n/a|n/a
              No|0|n/a|n/a
IPSecGWv6~7   1
              n/a
              IPSecGWv6~6
              ESP-Hmac_Md5-AES
              Y0
              Dynamic Anchor|n/a|Active|Permit|Routed|Outbound|Yes
              0|All|n/a
              TCP(6)|n/a|n/a|n/a|n/a
              n/a|None|Rule
              2001:db8:10::81:2:0
              112
              n/a
              Packet
              All|n/a|Rule
              2001:db8:10::81:8:1
              n/a
              2001:db8:10::81:8:6
              Packet
              All|n/a|Rule
              n/a|n/a|n/a
              2008/02/11 18:13:12|2008/02/11 18:13:12
              Silent|n/a|n/a|n/a|n/a
              n/a|n/a
              No|0|n/a|n/a
IPSecGWv6~7   2
              n/a
              IPSecGWv6~6
              ESP-Hmac_Md5-AES
              Y3
              Dynamic|n/a|Active|Permit|Routed|Inbound|Yes
              0|All|n/a
              TCP(6)|n/a|n/a|n/a|n/a
              n/a|None|Rule
              2001:db8:10::81:8:1
              n/a
              2001:db8:10::81:8:6
              Packet
              All|n/a|Rule
              2001:db8:10::81:2:0
              112
              n/a
              Packet
              All|n/a|Rule
              n/a|n/a|n/a
              n/a|n/a
              Silent|n/a|n/a|n/a|n/a
              n/a|n/a
              No|0|n/a|n/a
```

```
IPSecGWv6~7 |2
            |n/a
            |IPSecGWv6~6
            |ESP-Hmac_Md5-AES
            |Y0
            |Dynamic Anchor|n/a|Active|Permit|Routed|Inbound|Yes
            |0|All|n/a
            |TCP(6)|n/a|n/a|n/a|n/a
            |n/a|None|Rule
            |2001:db8:10::81:8:1
            |n/a
            |2001:db8:10::81:8:6
            |Packet
            |All|n/a|Rule
            |2001:db8:10::81:2:0
            |112
            |n/a
            |Packet
            |All|n/a|Rule
            |n/a|n/a|n/a
            |2008/02/11 18:13:12|2008/02/11 18:13:12
            |Silent|n/a|n/a|n/a|n/a
            |n/a|n/a
            |No|0|n/a|n/a

8 entries selected
```

detail

Displays IP security information in detail format. Detail format displays all applicable details for the selected IP security information. Each entry can span multiple lines or even multiple screens. Each field of each entry record is shown with both the heading and value for the field. Entry records are separated by a line of asterisks.

```
ipsec -p tcpcs4 -f display -r detail
```

```
CS V2R1 ipsec  Stack Name: TCPCS4  Tue Feb 14 06:54:24 2012
Primary:  Filter          Function: Display           Format:   Detail
Source:   Stack Policy    Scope:    Current           TotAvail: 164
Logging:  On              Predecap: Off               DVIPSec:  Yes
NatKeepAlive:  20         FIPS140:  No
Defensive Mode: Inactive

FilterName:                   IPSecGWv4~7
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv4~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y6
Type:                         Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
```

```
SourceAddress:                10.81.2.0
SourceAddressPrefix:          24
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.81.8.1
DestAddressPrefix:            n/a
DestAddressRange:             10.81.8.6
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   IPSecGWv4~7
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv4~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                10.81.2.0
SourceAddressPrefix:          24
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.81.8.1
DestAddressPrefix:            n/a
DestAddressRange:             10.81.8.6
DestAddressGranularity:       Packet
DestPort:                     All
```

```
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   IPSecGWv4~7
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         IPSecGWv4~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y6
Type:                         Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Inbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                10.81.8.1
SourceAddressPrefix:          n/a
SourceAddressRange:           10.81.8.6
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.81.2.0
DestAddressPrefix:            24
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
```

```
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   IPSecGWv4~7
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         IPSecGWv4~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Inbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                10.81.8.1
SourceAddressPrefix:          n/a
SourceAddressRange:           10.81.8.6
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.81.2.0
DestAddressPrefix:            24
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   IPSecGWv6~7
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv6~6
VpnActionName:                ESP-Hmac_Md5-AES
```

```
TunnelID:                     Y3
Type:                         Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                2001:db8:10::81:2:0
SourceAddressPrefix:          112
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  2001:db8:10::81:8:1
DestAddressPrefix:            n/a
DestAddressRange:             2001:db8:10::81:8:6
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
************************************************************************
FilterName:                   IPSecGWv6~7
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv6~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
```

```
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                2001:db8:10::81:2:0
SourceAddressPrefix:          112
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  2001:db8:10::81:8:1
DestAddressPrefix:            n/a
DestAddressRange:             2001:db8:10::81:8:6
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   IPSecGWv6~7
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         IPSecGWv6~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y3
Type:                         Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Inbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                2001:db8:10::81:8:1
SourceAddressPrefix:          n/a
SourceAddressRange:           2001:db8:10::81:8:6
SourceAddressGranularity:     Packet
SourcePort:                   All
```

```
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  2001:db8:10::81:2:0
DestAddressPrefix:            112
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
************************************************************************
FilterName:                   IPSecGWv6~7
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         IPSecGWv6~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Inbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                2001:db8:10::81:8:1
SourceAddressPrefix:          n/a
SourceAddressRange:           2001:db8:10::81:8:6
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  2001:db8:10::81:2:0
DestAddressPrefix:            112
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
```

```
CreateTime:                2012/02/14 18:13:12
UpdateTime:                2012/02/14 18:13:12
DiscardAction:             Silent
MIPv6Type:                 n/a
MIPv6TypeGranularity:      n/a
TypeRange:                 n/a
CodeRange:                 n/a
RemoteIdentityType:        n/a
RemoteIdentity:            n/a
FragmentsOnly:             No
FilterMatches:             0
LifetimeExpires:           n/a
AssociatedStackCount:      n/a
************************************************************************

8 entries selected
```

wide

Displays IP security information in wide format. Wide format displays each entry record (and the heading) on a single line of output. The heading line is first and each heading name is delimited by a vertical bar (|). This is followed by a line for each entry with all the data on a single line; values are also delimited by a vertical bar (|). Wide format is intended for use when redirecting output to a file. If this format is output to the screen, the lines wrap. See the following sample output for a key to the fields that are displayed.

ipsec -p tcpcs4 -f display -r wide

```
CS V2R1 ipsec  Stack Name: TCPCS4  Tue Feb 14 06:54:42 2012
Primary:  Filter          Function: Display            Format:   Wide
Source:   Stack Policy    Scope:    Current            TotAvail: 164
Logging:  On              Predecap: Off                DVIPSec:  Yes
NatKeepAlive:  20
Defensive Mode: Inactive
FilterName|FilterNameExtension|GroupName|LocalStartActionName|VpnActionName|Tunn
elID|Type|DefensiveType|State|Action|Scope|Direction|OnDemand|SecurityClass|Logg
ing|LogLimit|Protocol|ICMPType|ICMPTypeGran|ICMPCode|ICMPCodeGran|OSPFType|TCPQu
alifier|ProtocolGran|SourceAddress|SourceAddressPrefix|SourceAddressRange|Source
AddressGran|SourcePort|SourcePortRange|SourcePortGran|DestAddress|DestAddressPre
fix|DestAddressRange|DestAddressGran|DestPort|DestPortRange|DestPortGran|OrigRmt
ConnPort|RmtIDPayload|RmtUdpEncapPort|CreateTime|UpdateTime|DiscardAction|MIPv6T
ype|MIPv6TypeGran|TypeRange|CodeRange|RemoteIdentityType|RemoteIdentity|Fragment
sOnly|FilterMatches|LifetimeExpires|AssociatedStackCount

IPSecGWv4~7|1|n/a|IPSecGWv4~6|ESP-Hmac_Md5-AES|Y6|Dynamic|n/a|Active|Permit|Rout
ed|Outbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|10.81.2.0|24|n/a|
Packet|All|n/a|Rule|10.81.8.1|n/a|10.81.8.6|Packet|All|n/a|Rule|n/a|n/a|n/a|n/a|
n/a|Silent|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/a|n/a
IPSecGWv4~7|1|n/a|IPSecGWv4~6|ESP-Hmac_Md5-AES|Y0|Dynamic Anchor|n/a|Active|Perm
it|Routed|Outbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|10.81.2.0|
24|n/a|Packet|All|n/a|Rule|10.81.8.1|n/a|10.81.8.6|Packet|All|n/a|Rule|n/a|n/a|n
/a|2008/02/11 18:13:12|2008/02/11 18:13:12|Silent|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n
/a|n/a
IPSecGWv4~7|2|n/a|IPSecGWv4~6|ESP-Hmac_Md5-AES|Y6|Dynamic|n/a|Active|Permit|Rout
ed|Inbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|10.81.8.1|n/a|10.8
1.8.6|Packet|All|n/a|Rule|10.81.2.0|24 |n/a|Packet|All|n/a|Rule|n/a|n/a|n/a|n/a|
n/a|Silent|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/a|n/a
IPSecGWv4~7|2|n/a|IPSecGWv4~6|ESP-Hmac_Md5-AES|Y0|Dynamic Anchor|n/a|Active|Perm
it|Routed|Inbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|10.81.8.1|n
/a|10.81.8.6|Packet|All|n/a|Rule|10.81.2.0|24|n/a|Packet|All|n/a|Rule|n/a|n/a|n/
a|2008/02/11 18:13:12|2008/02/11 18:13:12|Silent|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/
a|n/a
IPSecGWv6~7|1|n/a|IPSecGWv6~6|ESP-Hmac_Md5-AES|Y3|Dynamic|n/a|Active|Permit|Rout
ed|Outbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|2001:db8:10::81:2
:0|112|n/a|Packet|All|n/a|Rule|2001:db8:10::81:8:1|n/a|2001:db8:10::81:8:6|Packe
```

```
t|All|n/a|Rule|n/a|n/a|n/a|n/a|n/a|Silent|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/a|n/a
IPSecGWv6~7|1|n/a|IPSecGWv6~6|ESP-Hmac_Md5-AES|Y0|Dynamic Anchor|n/a|Active|Perm
it|Routed|Outbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|2001:db8:1
0::81:2:0|112|n/a|Packet|All|n/a|Rule|2001:db8:10::81:8:1|n/a|2001:db8:10::81:8:
6|Packet|All|n/a|Rule|n/a|n/a|n/a|2008/02/11 18:13:12|2008/02/11 18:13:12|Silent
|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/a|n/a
IPSecGWv6~7|2|n/a|IPSecGWv6~6|ESP-Hmac_Md5-AES|Y3|Dynamic|n/a|Active|Permit|Rout
ed|Inbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|2001:db8:10::81:8:
1|n/a|2001:db8:10::81:8:6|Packet|All|n/a|Rule|2001:db8:10::81:2:0|112|n/a|Packet
|All|n/a|Rule|n/a|n/a|n/a|n/a|n/a|Silent|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/a|n/a
IPSecGWv6~7|2|n/a|IPSecGWv6~6|ESP-Hmac_Md5-AES|Y0|Dynamic Anchor|n/a|Active|Perm
it|Routed|Inbound|Yes|0|All|n/a|TCP(6)|n/a|n/a|n/a|n/a|n/a|None|Rule|2001:db8:10
::81:8:1|n/a|2001:db8:10::81:8:6|Packet|All|n/a|Rule|2001 :db8:10::81:2:0|112|n/
a|Packet|All|n/a|Rule|n/a|n/a|n/a|2008/02/11 18:13:12|2008/02/11 18:13:12|Silent
|n/a|n/a|n/a|n/a|n/a|n/a|No|0|n/a|n/a

8 entries selected
```

The ipsec command report heading:
All display reports from the **ipsec** command begin with several heading lines, which give general information related to the request. The first three heading lines and the final line, which include a selection count, exist in every report. Some reports might also have additional heading lines that contain information specific to the primary option.

Tip: When the **-z** option or the **-x** option is specified on the command, the stack name on the first line of the report is changed from `Stack Name` to `NSS Client Name`.

Heading example:

```
Line
1)  CS V2R1 ipsec  Stack Name: TCPCS4  Fri Nov 25 06:53:45 2011
2)  Primary:  Filter          Function: Display            Format:   Short
3)  Source:   Stack Policy    Scope:    Current            TotAvail: 164

4)  Logging:  On              Predecap: Off                DVIPSec:  Yes
5)  NatKeepAlive:  20         FIPS140:  No
6)  Defensive Mode: Inactive
7)  Exclusion Address: 9.1.1.1
```

The first heading line shows the following fields:

Stack Name
The stack name that the command is associated with. If global defensive filters are being displayed (**-F dis -G**) the command is not associated with a stack. The Stack Name value is *ALL*.

NSS Client Name
The name that is associated with the NSS client's stack.

<*timestamp*>
The date and time of the report.

The second heading line shows:

Primary
The primary option as indicated by the request. The possible values are `Filter`, `Defensive Filt`, `IKE tunnel`, `Dynamic tunnel`, `Manual tunnel`, `Interface`, `IP Traffic Test`, `NATT Port Trans`, `NSS Server`, or `Stack NSS`.

Function
The function option for any report is `display`. If the request is for IKE

tunnels with `cascade` (**-k dis -e**), then the function field displays `display (cascade)`. If the request is for shadow dynamic tunnels (**-y dis -s**), then the function field displays `display (shadows)`. If the request is for global defensive filters (**-F dis -G**), then the function field shows as `display (global)`.

Format
The report format as indicated by the request. The possible values are `detail`, `short`, or `wide`.

The third heading line shows:

Source
The source of the data in the report.

Data sources are:

- Stack: Data is from the IP stack.
- IKED: Data is from the IKE daemon.
- DMD: Data is from the Defense Manager daemon.

For the Filter (**-f**) and IP traffic test (**-t**) primary options, the source is one of the following value:

- Stack Profile: Data is from the default IP security filter policy that is specified in the IP stack's profile.
- Stack Policy: Data is from the IP security filter policy that is specified by the Policy Agent.

For the defensive filter (**-F**) primary option, the source is one of the following value:

- Stack: Data is from the IP stack.
- DMD: Data is from the Defense Manager daemon.

For the Network security server (**-x**) primary option, the source is the server (data is from the NSS server).

Scope The scope as indicated by the request.

- For the Filter (**-f**) primary option, the value is either current, policy, or profile (see “IP filter (-f) primary option” on page 738 for a discussion of the difference between policy and profile).
- For the IKE tunnel (**-k**) primary option, the value is current or all.
- For the Dynamic tunnel (**-y**) primary option, the value is current or all.
- For all other reports, the value is `n/a`.

TotAvail
The total number of items (filters or tunnel data) that are available from the stack. Depending on the selection criteria that is specified on the request, the report might not include all available entries. For example, a dynamic tunnel display for all tunnels (using the default Scope value of `current`) might format three tunnel entries, but the TotAvail field indicates the value 8. Reissuing the command with the Scope value `all` displays all eight tunnel entries and reveals that older, refreshed tunnels were not shown in the original display. For displays that are not stack oriented (Source is `IKED`), the value is `n/a`.

For the Filter (**-f**) and Defensive Filter (**-F**) primary options, the fourth heading line shows:

Logging

Indicates whether packet filter logging is in use globally for IP security filters.

- If the Source value is Stack Profile, the Logging value indicates the setting of the LOGENABLE or LOGDISABLE keyword of the IPSEC statement.
- If the Source value is Stack Policy, the Loggingvalue is the same as the FilterLogging setting on the IpFilterPolicy statement.
- If the Source value is Stack, the value is n/a.
- If the Source value is DMD, the value is n/a.

Tip: Packet filter logging is always in use for defensive filters at a global level. Each defensive filter indicates whether packet filtering is in use for that filter.

Predecap

Indicates whether decapsulated packets are first filtered at the stack.

- If the Source value is Stack Profile, the Predecap value is Off.
- If the Source value is Stack Policy, the value indicates the PreDecap setting of the IpFilterPolicy statement.
- If the Source value is Stack, the value is n/a.
- If the Source value is DMD, the value is n/a.

DVIPSec

Indicates whether the filters for IP security tunnels that are associated with dynamic VIPA addresses can be distributed or moved during VIPA takeover or giveback. The value indicates the setting of the DVIPSEC keyword of the IPSEC statement in the TCPIP profile. This value applies to the treatment of both Stack Profile filters and Stack Policy filters. If the Source value is Stack or DMD, the value is n/a.

For the IP traffic test (**-t**) primary option, the fourth heading line shows:

TestData

Shows the test data as indicated from the request. The first and second positional fields are the source and destination IP address, respectively. The third positional field is the specified protocol. If the protocol is TCP or UDP, then the fourth and fifth positional fields are the source and destination port numbers, respectively.

For the IKE network security (**-w**) primary option the fourth heading line shows:

System Name

The name of the system on which the IKE daemon is running.

For the Network security server (**-x**) primary option the fourth heading line shows:

System Name

The name of the system on which the NSS server is running.

For all other primary options, there is no fourth heading line.

For the Filter (**-f**) and Defensive Filter (**-F**) primary options, the fifth heading line shows:

NatKeepAlive

The NAT keep alive interval in seconds that was defined with the

NatKeepAliveInterval parameter on the KeyExchangePolicy statement. The value can be 0 (indicating that NAT keep alive messages should never be sent), or in the range 20 – 999 (indicating the number of seconds of inactivity that will trigger the sending of a NAT keep alive message). The default is 20 seconds. If the Source value is DMD, the value is `n/a`.

`FIPS140`
Specifies whether the stack is performing cryptographic operations using cryptographic algorithms and modules that are designed to meet the Federal Information Processing Standard (FIPS 140) security requirements. Possible values are:

Yes All cryptographic operations performed by the stack are designed to meet the FIPS 140 security requirements.

No The cryptographic operations performed by the stack are not designed to meet the FIPS 140 security requirements.

n/a On the -F display, the FIPS140 field contains the value of n/a.

For the Filter (**-f**) and Defensive Filter (**-F**) and IP traffic test (**-t**) primary options, subsequent heading lines show the following information:

`Defensive Mode`
Indicates the defensive filtering mode for the stack. The value is the same as the Mode setting on the DmStackConfig statement in the Defense Manager daemon (DMD) configuration file. The value is Active, Simulate, or Inactive. The value is Inactive if the Mode setting on the DmStackConfig statement is Inactive or if there is no DmStackConfig statement for this stack. If the Source value is DMD, the value is `n/a`.

`Exclusion Address`
If defensive filter processing is being used, you can specify an exclusion list of up to ten IP addresses or subnets in the DMD configuration file. This is intended to allow administrative access to the TCP/IP stack that could be inadvertently blocked by defensive filters. Inbound packets that originate from an IP address that is in the exclusion list are excluded from defensive filter processing. Outbound packets that are destined to an IP address that is in the exclusion list are excluded from defensive filter processing. Zero to ten Exclusion Address lines are included in the report heading.

The final line of any display report shows how many entries were actually listed in the report. Depending on the selection criteria that was specified on the request, the count of entries in the report might be less than the entire set.

The ipsec command report data:
All data fields are shown, even if some of the fields are not applicable to the type of entry that is being displayed or if some of the fields are not applicable to the context of the data. For example, if a filter protocol is AH, the fields labeled ICMPType and ICMPCode remain part of the display, even though their values are `n/a`.

IP filter (-f) primary option

The **-f** primary option is used to display and manage IP filter rules that are used in the TCP/IP stack. The current IP security filter rules can originate from static configuration in the TCPIP profile (referred to here as PROFILE) or indirectly from a variety of filter and tunnel specifications, which are managed through the Policy Agent (referred to here as POLICY). A display of the current filters includes both

IP security filter rules and defensive filter rules, if any exist. The defensive filter (**-F**) primary option provides management and further display of defensive filters.

See “The z/OS UNIX ipsec command IP filter (-f) option” on page 708 for parameter descriptions.

IP filter (-f) primary option syntax:
See “The z/OS UNIX ipsec command syntax” on page 700 for **-f** primary option syntax.

IP filter (-f) primary option command examples:

ipsec -f display -c current -p tcpcs1
: Displays the current filter rules from stack tcpcs1. Both IP security filters and defensive filters are included, if in use.

ipsec -f display -c profile
: Displays the profile IP security filter rules from the default stack.

ipsec -f dis -z nsclient1 -a y3
: Displays the current filter rules from client nsclient1 that are related to dynamic tunnel y3. The request is directed to the NSS server.

ipsec -f dis -p tcpcs1 -a y2 -h
: Displays the current filter rules from stack tcpcs1 that are related to dynamic tunnel y2 and include associated NRFs.

ipsec -f default -z nsclient1
: Changes the IP security filter rule set that was obtained through the Policy Agent to the default IP security filter policy that was specified in the TCPIP profile. The request is directed to the NSS server.

ipsec -f reload
: Changes the IP security filter rule set from the default IP security filter policy that was specified in the TCPIP profile to the IP security filter policy that was created in the Policy Agent.

IP filter (-f) primary option report examples:

`ipsec -p tcpcs -f display -a Y11`

```
CS V2R1 ipsec  Stack Name: TCPCS  Fri Nov 25 16:02:46 2011
Primary:  Filter          Function: Display            Format:   Detail
Source:   Stack Policy    Scope:    Current            TotAvail: 22
Logging:  On              Predecap: Off                DVIPSec:  No
NatKeepAlive:  0          FIPS140:  No
Defensive Mode: Inactive

FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y11
Type:                         NATT Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
```

```
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 10.37.55.212
RmtUdpEncapPort:              4500
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
************************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y0
Type:                         NATT Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
```

```
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          Rule
ICMPCode:                     n/a
ICMPCodeGranularity:          Rule
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
```

```
RmtUdpEncapPort:              n/a
CreateTime:                   2011/02/13 15:17:06
UpdateTime:                   2011/02/13 15:17:06
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         Rule
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                69
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
...

6 entries selected
```

ipsec -p tcpcs -f display -a y26 -h

```
CS V2R1 ipsec  Stack Name: TCPCS  Fri Nov 25 12:05:50 2011
Primary:  Filter          Function: Display            Format:   Detail
Source:   Stack Policy    Scope:    Current            TotAvail: 30
Logging:  On              Predecap: Off                DVIPSec:  No
NatKeepAlive:  0
Defensive Mode: Inactive
  FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y26
Type:                         NATT Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 10.37.55.211
RmtUdpEncapPort:              4500
```

```
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y26
Type:                         NRF
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          n/a
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   23
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     3755
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              3755
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
```

```
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y26
Type:                         NRF
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          n/a
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   623
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     40645
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              40645
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y0
Type:                         NATT Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
```

```
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                9.42.105.78
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.27.153.14
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                4
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
************************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          Rule
ICMPCode:                     n/a
ICMPCodeGranularity:          Rule
OSPFType:                     n/a
```

```
TCPQualifier:               n/a
ProtocolGranularity:        Rule
SourceAddress:              9.42.105.78
SourceAddressPrefix:        n/a
SourceAddressRange:         n/a
SourceAddressGranularity:   Packet
SourcePort:                 n/a
SourcePortRange:            n/a
SourcePortGranularity:      Rule
DestAddress:                9.27.153.14
DestAddressPrefix:          n/a
DestAddressRange:           n/a
DestAddressGranularity:     Packet
DestPort:                   n/a
DestPortRange:              n/a
DestPortGranularity:        Rule
OrigRmtConnPort:            n/a
RmtIDPayload:               n/a
RmtUdpEncapPort:            n/a
CreateTime:                 2011/02/12 12:54:48
UpdateTime:                 2011/02/12 12:54:48
DiscardAction:              Silent
MIPv6Type:                  n/a
MIPv6TypeGranularity:       Rule
TypeRange:                  n/a
CodeRange:                  n/a
RemoteIdentityType:         n/a
RemoteIdentity:             n/a
FragmentsOnly:              No
FilterMatches:              4
LifetimeExpires:            n/a
AssociatedStackCount:       n/a
************************************************************************

...

10 entries selected
```

Note: The inbound entries were truncated from the previous example. They have the same information format as the outbound entries that are displayed in that example.

IP filter (-f) primary option report field descriptions:
For more information about the header, see "The ipsec command report heading" on page 735.

FilterName
All filter rules have a base name that is used for reference purposes.

IP security filters
The base FilterName value is assigned by the system for filters that were created from the TCPIP profile. For filters that were created from POLICY, the base FilterName value corresponds to the **name** field of the IpFilterRule statement. This is the name to use when you specify **ipsec** command selection criteria using the **-n** option.

Defensive filters
The base FilterName value corresponds to the name that was specified with the **-N** *DefensiveFilterName* on the **ipsec -F add** command that created the filter. This is the name to use when you specify **ipsec** command selection criteria using the **-N** option. For global defensive filters, the generated stack specific filters have the same FilterName value as the global filters.

FilterNameExtension

Base filters as defined by the administrator might result in multiple filter rules maintained in the stack. The FilterNameExtension value is the system-assigned value that (when combined with the FilterName value) makes the filter unique.

GroupName

In POLICY, individual IpFilterRule statements can be grouped together into an IpFilterGroup group, which carries a name. If the individual filter is defined to an IpFilterGroup group, that name is displayed in this field. Use this name when specifying **ipsec** command selection criteria using the **-g** option.

LocalStartActionName

In POLICY, the IpFilterRule statement can reference an IpLocalStartAction statement (as part of the IpDynVpnAction specification) in order to control the local activation of a dynamic tunnel. If the individual filter is associated with an IpLocalStartAction statement, that name is displayed in this field.

VpnActionName

In POLICY, the IpFilterRule statement can reference an IpManVpnAction specification (for manual tunnels) or an IpDynVpnAction specification (for dynamic tunnels) in order to define how traffic should be managed between two security endpoints. If the individual filter is associated with an IpManVpnAction or IpDynVpnAction specification, that name is displayed in this field.

TunnelID

If the filter was created to control data traffic for a manual or dynamic tunnel, the tunnel ID with which the filter is associated is displayed in this field. The TunnelID parameter has the value `M` (for manual) or `Y` (for dynamic) followed by an arbitrary positive integer that was assigned by the system when the tunnel was activated. Use this name when specifying an **ipsec** command selection criteria using the **-a** option. If the filter **Type** is `Dynamic Anchor` or `NATT anchor`, the **TunnelID** is `Y0`. If the filter is not associated with a tunnel, the value is `n/a`.

Type This field indicates whether the filter entry is one of the following values:

Manual

Statically defined for a manual tunnel.

Dynamic Anchor

Statically defined to control the creation of new filters for a dynamic tunnel.

Dynamic

Dynamically defined through negotiation through an IKE exchange.

NATT Dynamic

Dynamically defined through negotiation through an IKE exchange. The NATT dynamic filter is defined for an SA that traverses a NAT in certain configurations. See NATT anchor and NATT dynamic filters in the z/OS Communications Server: IP Configuration Guide for more information.

NATT Anchor

Dynamically defined to anchor the NATT dynamics.

NRF Dynamically defined on inbound packet processing. The NRF filter is defined for traffic that is received over an SA that traverses a NAT in certain configurations. See NAT Resolution Filters (NRF) in the z/OS Communications Server: IP Configuration Guide for more information.

Generic
Statically defined to control traffic other than for a manual or dynamic tunnel

Defensive
Defensive filter defined with an **ipsec -F add** command.

DefensiveType
Indicates whether the defensive filter was added only to this stack or as a global filter to all eligible stacks on the z/OS system. Possible values are Global or Stack. If the DefensiveType value is Global, the filter was added globally to the z/OS system with the **-G** option. If the DefensiveType value is Stack, the filter was added to this stack with the **-p** option. If the filter is not a defensive filter, the value is `n/a`.

State The current state of the filter entry. The **ipsec** command always displays active filter entries, so the value is always `Active`.

Action
Indicates the action that is to be taken on data traffic when the filter entry is invoked. Possible values are Permit, Deny, Defensive Block, or Defensive Simulate.

Results:

- When data traffic is to be protected with IPSec, the **Action** field displays the value `Permit` and the **Type** field displays one of the following values: `Dynamic, Manual, NATT Dynamic, Dynamic Anchor, or NATT Anchor`.
- A defensive filter has the action Defensive Block or Defensive Simulate that indicates the mode of the filter. If the Simulate mode is configured on the DmStackConfig statement in the DMD configuration file, that value overrides a mode of block on the defensive filter. If the Active mode is coded on the DmStackConfig statement, the mode setting of the filter is used. The Defensive Mode field in the report heading displays the mode from the configuration file of the DMD.

Scope Indicates the scope of data traffic that is encompassed by the filter entry. Possible values are `Local`, `Routed`, or `Both`.

Direction
Indicates the direction of data traffic to which the filter entry applies. Possible values are `Outbound` or `Inbound`.

OnDemand
Indicates whether the filter entry was created to handle on-demand data traffic. This field is applicable only to filters associated with dynamic tunnels.

SecurityClass
The security class to which the filter entry applies. The security class is used to group interfaces by secure traffic patterns. The value 0 indicates that all security classes apply. For policy-configured dynamic anchor filters and generated dynamic filter rules, the security class value is always 0. For defensive filters, the security class value is always 0.

Logging

Indicates the logging that is to be performed when the filter is invoked. Possible values are:

All A log entry is generated if data traffic is permitted or denied.

Permit
A log entry is generated only when data traffic is permitted.

Deny A log entry is generated only when data traffic is denied.

None No log entries are generated from this filter.

LogLimit

Indicates the average rate of defensive filter match messages that are allowed in a five-minute interval. A value of `0` indicates that filter match messages are not limited for this defensive filter. If the filter is not a defensive filter, the value is `n/a`. For more information see filter-match logging in z/OS Communications Server: IP Configuration Guide.

DiscardAction

Indicates the discard action for packets that are discarded as a result of this filter rule. Possible values are:

Silent Packets are discarded silently.

ICMP When a packet is discarded, an ICMP or ICMPv6 error is sent to the origin of the discarded packet to indicate that the packet was administratively prohibited.

Protocol

Indicates the protocol to which the filter applies. If the specification is for a specific protocol, the name and number of the protocol is displayed for commonly used protocols, or if the protocol is not commonly used, only the number of the protocol is displayed. See the following possible values:

All The specification is for all protocols.

Opaque
The specification is for packets whose protocol is indeterminate.

ICMPType

Indicates the ICMP type of the data traffic. The filter will match this type when the protocol is ICMP or ICMPv6. The value `All` indicates that the filter matches all ICMP types. The value `Opaque` indicates that the filter matches only packets with unknown ICMP types caused by packet fragmentation. The value `n/a` indicates that this field is not applicable to the current protocol.

ICMPTypeGranularity

Granularity values are set in an IpLocalStartAction statement. These values control the data elements that are used in negotiating dynamic tunnels. If the granularity value was not set by an associated IpLocalStartAction statement, the display shows the default value. The value `Rule` indicates that dynamic tunnel negotiation uses the ICMP Type specification from the matching filter rule. The value `Packet` indicates that the dynamic tunnel negotiation uses the ICMP Type specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter type is `Dynamic Anchor` and the protocol is `All`, `ICMP`, or `ICMPv6`; the value `n/a` is displayed for all other cases.

ICMPCode

Indicates the ICMP code of the data traffic. The filter matches this code when the protocol is ICMP or ICMPv6. The value `All` indicates that the filter matches all ICMP codes. The value `Opaque` indicates that the filter matches only packets with unknown ICMP codes caused by fragmentation. The value `n/a` indicates that this field is not applicable to the current protocol.

ICMPCodeGranularity
Granularity values are set from an IpLocalStartAction statement. These values control the data elements that are used in negotiating dynamic tunnels. If the granularity value was not set by an associated IpLocalStartAction statement, the display shows the default value. The value `Rule` indicates that dynamic tunnel negotiation uses the ICMP Code specification from the matching filter rule. The value `Packet` indicates that the dynamic tunnel negotiation uses the ICMP Code specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter type is `Dynamic Anchor` and the protocol is `All`, `ICMP`, or `ICMPV6`; the value `n/a` is displayed for all other cases.

OSPFType
Indicates the OSPF type of the data traffic. The filter matches this type when the protocol is OSPF. The value `All` indicates that the filter matches all OSPF types. The value `n/a` indicates that this field is not applicable to the current protocol.

MIPv6Type
Indicates the MIPv6 type of the data traffic that the filter matches when the protocol is MIPv6 . The value `All` indicates that the filter matches all MIPv6 types. The value `Opaque` indicates that the filter matches only packets with unknown MIPv6 types caused by packet fragmentation . The value `n/a` indicates that this field is not applicable to the current protocol.

MIPv6TypeGranularity
Granularity values are set in an IpLocalStartAction statement. These values control the data elements that are used in negotiating dynamic tunnels. If the granularity value was not set by an associated IpLocalStartAction statement, the display shows the default value. The value `Rule` indicates that dynamic tunnel negotiation uses the MIPv6 type specification from the matching filter rule. The value `Packet` indicates that the dynamic tunnel negotiation uses the MIPv6 type specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter type is `Dynamic Anchor` and the protocol is `All` or `MIPV6`; the value `n/a` is displayed for all other cases.

TypeRange
Indicates the upper value in the range of type numbers of the data traffic that the filter matches when the protocol is ICMP, ICMPv6, or MIPv6. The value `n/a` indicates that this field is not applicable to the current protocol, or that the tunnel's type selectors are entirely specified by either the ICMPType value or the MIPv6Type value.

CodeRange
Indicates the upper value in the range of code numbers of the data traffic that the filter matches when the protocol is ICMP or ICMPv6. The value `n/a` indicates that this field is not applicable to the current protocol, or that the tunnel's code selectors are entirely specified by the Code value.

TCPQualifier
If the protocol is TCP and the direction of the filter rule specification was bidirectional, then additional criteria might have been specified to control

TCP connection traffic. If the filter being displayed indicates a Direction value of `Outbound`, then this field displays `Connect Outbound` to indicate that TCP outbound connects are being controlled. If the filter being displayed indicates a Direction value of `Inbound`, this field displays `Connect Inbound` to indicate that TCP inbound connects are being controlled. If the protocol is not TCP, this field is not applicable.

ProtocolGranularity
Granularity values are set from an IpLocalStartAction statement to control the data elements used in negotiating dynamic tunnels. If the granularity value was not set by an associated IpLocalStartAction statement, the display shows the default value. The value `Rule` indicates that dynamic tunnel negotiation uses the protocol specification from the matching filter rule. The value `Packet` indicates that the dynamic tunnel negotiation uses the protocol specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter Type is Dynamic Anchor. The value `n/a` is displayed for all other cases.

SourceAddress
The source IP address to which this filter applies.

- If the SourceAddressPrefix and SourceAddressRange fields both indicate the value `n/a`, then the filter applies to this single IP address. Otherwise, the SourceAddress value is the base IP address for a collection of addresses to which the filter applies.
- If the SourceAddressPrefix field has a value, it represents a subnet mask and the combination of the SourceAddress value and the subnet mask defines the collection of addresses to which this filter applies.
- If the SourceAddressRange field has a value, it is the high end of a range of IP addresses (inclusive) to which this filter applies.
- If the SourceAddress value is all zeroes and either the SourceAddressPrefix value is 0 or the SourceAddressRange value is 255.255.255.255, then the filter applies to all source IP addresses.

SourceAddressPrefix
If this field contains a value, it represents a subnet mask, which in combination with the SourceAddress value, defines a collection of addresses to which this filter applies. The SourceAddressPrefix field is an integer that defines the number of high-order bits to be interpreted as a subnet mask. For example, the SourceAddressPrefix value 24 defines a subnet mask of 24 high-order bits or an address of 255.255.255.0. This subnet, as applied to the base IP address (the value SourceAddress), is the collection of addresses to which the filter applies.

SourceAddressRange
If this field contains a value, then the SourceAddress value is the first address of the range and the SourceAddressRange value indicates the final address of the range in a collection of addresses (inclusive) to which the filter applies.

SourceAddressGranularity
Granularity values are set from an IpLocalStartAction statement to control the data elements used in negotiating dynamic tunnels. If granularity values are not set by an associated IpLocalStartAction statement, the display shows the default value. The value `Rule` indicates that dynamic tunnel negotiation uses the source IP address specification from the matching filter rule. The value `Packet` indicates that dynamic tunnel negotiation uses the source IP address specification from the packet that

initiated the dynamic tunnel activation. This field applies only when the filter Type is Dynamic Anchor. The value n/a is displayed for all other cases.

SourcePort

Indicates the source port number of the data traffic that the filter matches when the protocol is TCP or UDP. The value All indicates that the filter matches all source ports. The value Opaque indicates that the filter matches only packets with unknown source ports caused by packet fragmentation. The value n/a indicates that this field is not applicable to the current protocol.

SourcePortRange

Indicates the upper value in the range of source port numbers of the data traffic that the filter matches when the protocol is TCP or UDP. The value n/a indicates that this field is not applicable to the current protocol, or that the filter's source port selectors are entirely specified by the SourcePort value.

SourcePortGranularity

Granularity values are set from an IpLocalStartAction statement to control the data elements used in negotiating dynamic tunnels. If granularity values are not set by an associated IpLocalStartAction statement, the display shows the default value. The value Rule indicates that dynamic tunnel negotiation uses the source port specification from the matching filter rule. The value Packet indicates that dynamic tunnel negotiation uses the source port specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter Type is Dynamic Anchor and Protocol is All, TCP or UDP. The value n/a is displayed for all other cases.

DestAddress

The destination IP address to which this filter applies.

- If the DestAddressPrefix and DestAddressRange fields both indicate the value n/a, then the filter applies to this single IP address. Otherwise, the DestAddress value is the base IP address for a collection of addresses to which the filter applies.
- If the DestAddressPrefix field has a value, it represents a subnet mask and the combination of the DestAddress value and the subnet mask defines the collection of addresses to which this filter applies.
- If the DestAddressRange field has a value, it is the high end of a range of IP addresses (inclusive) to which this filter applies.
- If the DestAddress value is all zeroes and either the DestAddressPrefix is 0 or the DestAddressRange is 255.255.255.255, then the filter applies to all destination IP addresses.

DestAddressPrefix

If this field contains a value, it represents a subnet mask, which in combination with the DestAddress value, defines a collection of addresses to which this filter applies. The DestAddressPrefix value is an integer that defines the number of high-order bits to be interpreted as a subnet mask. For example, if the DestAddressPrefix field contains the value 24, this defines a subnet mask of 24 high-order bits, or 255.255.255.0. This subnet, as applied to the base IP address (the value of DestAddress), is the collection of addresses to which the filter applies.

DestAddressRange

If this field contains a value, then the DestAddress value is the first address of the range and this field indicates the final address of the range in a collection of addresses (inclusive) to which the filter applies.

DestAddressGranularity

Granularity values are set from an IpLocalStartAction statement to control the data elements used in negotiating dynamic tunnels. If granularity values are not set by an associated IpLocalStartAction statement, the display shows the default value. The value `Rule` indicates that dynamic tunnel negotiation uses the destination IP address specification from the matching filter rule. The value `Packet` indicates that dynamic tunnel negotiation uses the destination IP address specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter Type is Dynamic Anchor. The value `n/a` is displayed for all other cases.

DestPort

Indicates the destination port number of the data traffic that the filter matches when the protocol is TCP or UDP. The value `All` indicates that the filter matches all destination ports. The value `Opaque` indicates that the filter matches only packets with unknown destination ports caused by packet fragmentation. The value `n/a` indicates that this field is not applicable to the current protocol.

DestPortRange

Indicates the upper value in the range of destination port numbers of the data traffic that the filter matches when the protocol is TCP or UDP. The value `n/a` indicates that this field is not applicable to the current protocol, or that the filter's destination port selectors are entirely specified by the DestPort value.

DestPortGranularity

Granularity values are set from an IpLocalStartAction statement to control the data elements used in negotiating dynamic tunnels. If granularity values are not set by an associated IpLocalStartAction statement, the display shows the default value. A value of `Rule` indicates that dynamic tunnel negotiation will use the destination port specification from the matching filter rule. A value of `Packet` indicates that dynamic tunnel negotiation will use the destination port specification from the packet that initiated the dynamic tunnel activation. This field applies only when the filter Type is Dynamic Anchor and Protocol is `All`, `TCP` or `UDP`. The value `n/a` is displayed for all other cases.

RemoteIdentityType

Specifies the type of the remote identity. This field is applicable only to dynamic anchor filters, dynamic filters, and NATT dynamic filters that are filtering on the basis of remote identity. If a value is specified in this field, it represents the remote IKE identity that is coded on a dynamic anchor filter, or the actual remote IKE identity determined for a dynamic filter or NATT dynamic filter. This field does not contain a value for shadow filters. Possible values are:

ID_IPV4_ADDR

An IPv4 address.

ID_IPV6_ADDR

An IPv6 address.

ID_FQDN

A fully qualified domain name.

ID_KEY_ID
An opaque byte stream.

ID_USER_FQDN
A user at a fully qualified domain name.

ID_DER_ASN1_DN
An X.500 distinguished name.

n/a No remote identity is associated with this filter.

RemoteIdentity
Specifies the value of the remote identity. Contains the value `n/a` if no remote identity is present.

Restriction: If the RemoteIdentityType value is ID_KEY_ID, the RemoteIdentity value is truncated to avoid spanning multiple lines on a typical display device. An ellipsis is appended to the value to indicate that it was truncated. To display the entire value, use the wide (-r wide) display format.

FragmentsOnly
For a filter rule that might match routed traffic, this field indicates whether this filter rule applies to fragmented packets. Possible values are:

Yes The filter rule matches only fragmented packets.

No The filter rule matches both fragmented and non-fragmented packets.

OrigRmtConnPort
The connection port value of the remote endpoint. This field is applicable only for NRF filter entries. When the value is nonzero, remote port translation is being done and a remote port value of the connection might have been translated. The translated value is displayed as the SourcePort value for an inbound NRF entry and as the DestPort value for an outbound NRF entry. For other types of filter entries, the value in this field is `n/a`.

RmtIDPayload
For a NATT dynamic filter entry, this field displays the remote IP ID payload value for IKEv1 or remote Traffic Selector payload value for IKEv2. This field can display the value `none`, a single IP address, an IP address range, an IP address mask, or an MD5 hash of a non-IPv4 ID payload. For other types of filter entries, the value in this field is `n/a`.

RmtUdpEncapPort
For a NATT dynamic filter entry, this field is the UDP-encapsulated port number used by the remote security endpoint. For other types of filter entries, the value in this field is `n/a`.

CreateTime
For a statically defined filter that originates in the Policy Agent configuration, this field represents the time that the filter was first defined to the current instance of the TCP/IP stack. For a filter that originates in the TCP/IP profile, this field represents the time that the profile filter configuration was last replaced. For all dynamically defined filters, the value in this field is `n/a`.

UpdateTime
For a statically defined filter that originates in the Policy Agent configuration, this field represents the time that the attributes of the filter

were last updated in the current instance of the TCP/IP stack. For a filter that originates in the TCP/IP profile, this field represents the time that the profile filter configuration was last replaced. For all dynamically defined filters, the value in this field is `n/a`.

FilterMatches
The number of times that a packet has matched this filter. For a NATT Dynamic filter entry, the value in this field is 0.

LifetimeExpires
For a defensive filter, this is the time at which the filter expires. For other types of filter entries, the value of this field is `n/a`.

AssociatedStackCount
The value of this field is always n/a on this display. This field is applicable only for a global defensive filter that is displayed from the DMD with the **-G** option.

Defensive filter (-F) primary option

Use the **-F** primary option to display and manage defensive filters both in the TCP/IP stack and at a global z/OS system level. Defensive filters deny traffic or simulate a denial. Traffic is checked first against defensive filters. If the traffic is not denied by a defensive filter, then the IP security filters are checked. See defensive filtering information in z/OS Communications Server: IP Configuration Guide for details.

See "The z/OS UNIX ipsec command defensive filter (-F) option" on page 711 for parameter descriptions.

Defensive filter (-F) primary option syntax:
See "The z/OS UNIX ipsec command syntax" on page 700 for **-F** primary option syntax.

Defensive filter (-F) primary option command examples:

ipsec -F add srcip 192.30.30.0/24 dir inbound lifetime 30 mode block -p TCPCS1 -N Block_malformed
Adds a stack-specific defensive filter to the TCPCS1 stack with the name Block_malformed. The filter blocks inbound traffic from subnet 192.30.30.0/24 and remains installed in the stack for 30 minutes.

ipsec -F add destport 21 dir inbound lifetime 5 -G -N G_Block_local_FTP
Adds a global defensive filter with the name G_Block_local_FTP that blocks inbound traffic to port 21 for 5 minutes. The Defense Manager daemon (DMD) maintains a copy of the global filter and generates a copy that is installed in each local TCP/IP stack that is listed in the DMD configuration file that supports IP security.

ipsec -F update lifetime 10 -p TCPCS1 -N Block_malformed
Updates the lifetime for the stack-specific filter with the name Block_malformed on the TCPCS1 stack. The lifetime of the filter is 10 minutes from the time that the update is processed. The original lifetime value could be lengthened or shortened depending on when the update is processed.

ipsec -F delete -N G_Block_local_FTP -p TCPCS1
Deletes global filter G_Block_local_FTP from the TCPCS1 stack. The global filter remains in the DMD and it remains installed on other stacks on the local system.

ipsec -F display -p TCPCS1
Displays the defensive filters from stack TCPCS1.

ipsec -F display -G
Displays the global defensive filters from the DMD.

Defensive filter (-F) primary option report example:

```
ipsec -F dis -p tcpcs1 -N Block_malformed

CS V2R1 ipsec  Stack Name: TCPCS1    Tue Feb 14 16:08:56 2012
Primary:  Defensive Filt  Function: Display            Format:   Detail
Source:   Stack           Scope:    n/a                TotAvail: 32
Logging:  n/a             Predecap: n/a                DVIPSec:  n/a
NatKeepAlive:  20
Defensive Mode: Active
Exclusion Address: 9.1.1.1
Exclusion Address: 9.1.1.2

FilterName:                   Block_malformed
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                n/a
TunnelID:                     n/a
Type:                         Defensive
DefensiveType:                Stack
State:                        Active
Action:                       Defensive Block
Scope:                        Local
Direction:                    Inbound
OnDemand:                     n/a
SecurityClass:                0
Logging:                      All
LogLimit:                     100
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                192.30.30.0
SourceAddressPrefix:          24
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  0.0.0.0
DestAddressPrefix:            0
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 16:06:21
UpdateTime:                   2012/02/14 16:06:21
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
```

```
RemoteIdentityType:          n/a
RemoteIdentity:              n/a
FragmentsOnly:               No
FilterMatches:               0
LifetimeExpires:             2012/02/14 16:36:21
AssociatedStackCount:        n/a
*******************************************************************

1 entries selected
```

Defensive filter (-F) primary option field descriptions:
For more information about the header, see "The ipsec command report heading" on page 735.

FilterName
All filter rules have a base name that is used for reference purposes. For defensive filters, the base FilterName value corresponds to the name specified with the **-N** *DefensiveFilterName* on the **ipsec -F add** command that created the filter. This is the name to use when you specify **ipsec** command selection criteria using the **-N** option. For global defensive filters, the stack-specific filters that are generated have the same FilterName value as the global filters.

FilterNameExtension
Base filters as defined by the administrator might result in multiple filter rules being maintained in the stack. The FilterNameExtension value is the system-assigned value that (when combined with the FilterName value) makes the filter unique.

GroupName
Not applicable for defensive filters.

LocalStartActionName
Not applicable for defensive filters.

VpnActionName
Not applicable for defensive filters.

TunnelID
Not applicable for defensive filters.

Type The type value is always Defensive.

DefensiveType
Indicates whether the defensive filter was added only to this stack or as a global filter to all eligible stacks on the z/OS system. Possible values are Global or Stack. If the DefensiveType value is Global, the filter was added globally with the **-G** option to the z/OS system. If the DefensiveType value is Stack, the filter was added to this stack with the **-p** option.

State The current state of the filter entry. The possible values are Active or Pending Inactive. The **ipsec** command always displays active filter entries from the stack, so the value for these entries is always Active. For a global filter that is displayed from the DMD with the **-G** option, the State value is Pending Inactive when the global filter has expired but one or more of its stack specific copies is active. The global filter is retained to allow global update and delete operations.

Action
Indicates the action that is to be taken on data traffic when the filter entry is invoked. Possible values are Defensive Block or Defensive Simulate, whichever is the mode of the filter. If you code the mode Simulate on the

DmStackConfig statement in the DMD configuration file, this mode value overrides the defensive filter mode setting block. The mode value Active on the DmStackConfig statement honors the filter's mode setting. The Defensive Mode field in the report heading displays the mode from the DMD configuration file.

Scope Indicates the scope of data traffic that is encompassed by the filter entry. Possible values are Local, Routed, or Both.

Direction
Indicates the direction of data traffic to which the filter entry applies. Possible values are Outbound or Inbound.

OnDemand
Not applicable for defensive filters.

SecurityClass
The security class to which the filter entry applies. The security class is used to group interfaces by secure traffic patterns. The value 0 indicates that all security classes apply. For defensive filters, the security class value is always 0.

Logging
For a defensive filter with the Action value Defensive Block, the Logging field indicates the logging that is to be performed when the filter is invoked. Possible values are:

All A log entry is generated.

None No log entries are generated from this filter.

For a defensive filter with the Action value Defensive Simulate, logging is always performed when the filter is invoked. The Logging value is All (a log entry is generated).

LogLimit
Indicates the average rate of defensive filter match messages that are allowed in a five-minute interval. A value of `0` indicates that filter match messages are not limited for this defensive filter. For a global filter that is displayed from the DMD with the **-G** option, this field has the value **Unspecified** if loglimit was not specified on the ipsec add or update command. For more information, see filter-match logging in z/OS Communications Server: IP Configuration Guide.

Protocol
Indicates the protocol to which the filter applies. The value All indicates that the specification is for all protocols. If the specification is for a specific protocol, the name and number of the protocol is displayed for commonly used protocols. If the protocol is not commonly used, only the number of the protocol is displayed.

ICMPType
If the protocol is ICMP or ICMPv6, this field displays the type of ICMP message that is specified. If no ICMP type is specified, the value is All. If the protocol is not ICMP, this field is not applicable.

ICMPTypeGranularity
Not applicable for defensive filters.

ICMPCode
If the protocol is ICMP or ICMPv6, this field displays the code of ICMP

message that is specified. If no ICMP code is specified, the value is All. If the protocol is not ICMP, this field is not applicable.

ICMPCodeGranularity
Not applicable for defensive filters.

OSPFType
If the protocol is OSPF, the value is All. If the protocol is not OSPF, this field is not applicable.

TCPQualifier
Not applicable for defensive filters.

ProtocolGranularity
Not applicable for defensive filters.

SourceAddress
The source IP address to which this filter applies.

- If the SourceAddressPrefix and SourceAddressRange fields both contain the value `n/a`, then the filter applies to this single IP address. Otherwise, the SourceAddress value is the base IP address for a collection of addresses to which the filter applies.
- If the SourceAddressPrefix field has a value, the value is an integer that specifies the number of high-order bits that are to be interpreted as a subnet mask, which is combined with the SourceAddress value to define the collection of addresses to which this filter applies.
- If the SourceAddress value is 0.0.0.0 and the SourceAddressPrefix is 0, then the filter applies to all source IPv4 addresses.
- If the SourceAddress value is :: and the SourceAddressPrefix is 0, then the filter applies to all source IPv6 addresses.
- If the SourceAddress value is All for a global filter that is displayed from the DMD with the **-G** option, then the global filter applies to all IPv4 and IPv6 source addresses. For a stack that supports IPv6, the DMD installs both an IPv4 and IPv6 defensive filter in the stack. If the SourceAddress value is All, the DestAddress value is All.

SourceAddressPrefix
If this field contains a value, it is an integer that specifies the number of high-order bits that are to be interpreted as a subnet mask. This value , when combined with the SourceAddress value, defines a collection of addresses to which this filter applies. For example, the SourceAddressPrefix value 24 defines a subnet mask of 24 high-order bits or the address 255.255.255.0. This subnet, as applied to the base IP address (the value of SourceAddress), is the collection of addresses to which the filter applies.

SourceAddressRange
Not applicable for defensive filters.

SourceAddressGranularity
Not applicable for defensive filters.

SourcePort
This field contains the number of the source port to which this filter applies. If a port range is specified, then SourcePort value is the first port number of the range. If the filter applies to all source ports, the value is All.

SourcePortRange
If this field contains a value, then the SourcePort value is the first port

number of the range and this field indicates the final port number of the range in a collection of port numbers (inclusive) to which the filter applies.

SourcePortGranularity
Not applicable for defensive filters.

DestAddress
The destination IP address to which this filter applies.

- If the DestAddressPrefix and DestAddressRange fields both indicate the value n/a, then the filter applies to this single IP address. Otherwise, the DestAddress value is the base IP address for a collection of addresses to which the filter applies.
- If the DestAddressPrefix field has a value, that value represents a subnet mask. The combination of the DestAddress value and the subnet mask defines the collection of addresses to which this filter applies.
- If the DestAddress value is 0.0.0.0 and the DestAddressPrefix is 0, then the filter applies to all destination IPv4 addresses.
- If the DestAddress value is :: and the DestAddressPrefix is 0, then the filter applies to all destination IPv6 addresses.
- If the DestAddress value is All for a global filter displayed from the DMD with the **-G** option, then the global filter applies to all IPv4 and IPv6 destination addresses. For a stack that supports IPv6, the DMD installs both an IPv4 and IPv6 defensive filter in the stack. If the DestAddress value is All, the SourceAddress value is All.

DestAddressPrefix
If this field contains a value, the value represents a subnet mask, which in combination with the DestAddress value, defines a collection of addresses to which this filter applies. The DestAddressPrefix value is an integer that defines the number of high-order bits to be interpreted as a subnet mask. For example, if the DestAddressPrefix field contains the value 24, this defines a subnet mask of 24 high-order bits, or 255.255.255.0. This subnet, as applied to the base IP address (the value of DestAddress), is the collection of addresses to which the filter applies.

DestAddressRange
Not applicable for defensive filters.

DestAddressGranularity
Not applicable for defensive filters.

DestPort
Contains the number of the destination port to which this filter applies. If a port range was specified, then the DestPort value is the first port number of the range. If the filter applies to all destination ports, the value is All.

DestPortRange
If this field contains a value, then the DestPort value is the first port number of the range and this field indicates the final port number of the range in a collection of port numbers (inclusive) to which the filter applies.

DestPortGranularity
Not applicable for defensive filters.

OrigRmtConnPort
Not applicable for defensive filters.

RmtIDPayload
Not applicable for defensive filters.

RmtUdpEncapPort
Not applicable for defensive filters.

CreateTime
The time the defensive filter was added.

UpdateTime
The time the defensive filter was last updated.

DiscardAction
Indicates the discard action for packets that were discarded as a result of this filter rule. The value is Silent for a defensive filter, which indicates that packets are discarded silently with no ICMP notification.

MIPv6Type
Not applicable for defensive filters.

MIPv6TypeGranularity
Not applicable for defensive filters.

TypeRange
Not applicable for defensive filters.

CodeRange
Not applicable for defensive filters.

RemoteIdentityType
Not applicable for defensive filters.

RemoteIdentity
Not applicable for defensive filters.

FragmentsOnly
For a filter rule that might match routed traffic, this field indicates whether this filter rule applies to fragmented packets. Possible values are Yes (the rule matches only fragmented packets), No (the rule matches both fragmented and non-fragmented packets).

FilterMatches
The number of times a packet has matched this filter. For a global filter that is displayed from the DMD with the **-G** option, this field has the value `n/a`.

LifetimeExpires
The time the defensive filter will expire. For a global filter that is displayed from the DMD with the **-G** option, this field value is Expired when the global filter has expired but one or more of its stack-specific copies are still Active. The global filter is retained to allow global update and delete operations.

AssociatedStackCount
The number of TCP/IP stacks with which a global defensive filter is associated. This field is applicable only for a global filter that is displayed from the DMD with the **-G** option. For a defensive filter that is displayed without the **-G** option, this field has the value `n/a`.

Manual tunnel (-m) primary option

The **-m** primary option is used to display and manage manual tunnels as they are defined to the TCP/IP stack. Configuration for manual tunnels originates with IpFilterRule policy statements that include or reference IpManVpnAction statements.

See “The z/OS UNIX ipsec command manual tunnel (-m) option” on page 717 for parameter descriptions.

Manual tunnel (-m) primary option syntax:
For **-m** primary option syntax see “The z/OS UNIX ipsec command syntax” on page 700.

Manual tunnel (-m) primary option command examples:

ipsec -m display
Displays the current manual tunnel data from the default stack.

ipsec -m activate -z nsclient1 -n ipManVpnAct1
Activates the manual tunnel that was defined in policy as ipManVpnAct1 for client nsclient1. The request is directed to the NSS server.

ipsec -m deactivate -a M05
Deactivates the manual tunnel with an ID of M05 (the tunnel ID was found from an earlier display command) for the default stack.

Manual tunnel (-m) primary option report example:

```
ipsec -p tcpcs2 -m display

CS V2R1 ipsec  Stack Name: TCPCS2  Fri Nov 25 07:11:45 2011
Primary:  Manual tunnel   Function: Display           Format:   Detail
Source:   Stack           Scope:    Current           TotAvail: 2

TunnelID:                     M1
VpnActionName:                ManualTunnel-AH-SHA-AES~MVPN3
State:                        Active
HowToEncap:                   Tunnel
LocalEndPoint:                10.81.2.10
RemoteEndPoint:               10.81.8.10
HowToAuth:                    AH
 AuthAlgorithm:               HMAC-SHA1
 AuthInboundSpi:              7777        (0x     1E61)
 AuthOutboundSpi:             5555        (0x     15B3)
HowToEncrypt:                 AES-CBC
 KeyLength:                   128
 EncryptInboundSpi:           6666        (0x     1A0A)
 EncryptOutboundSpi:          4444        (0x     115C)
OutboundPackets:              0
OutboundBytes:                0
InboundPackets:               0
InboundBytes:                 0
PassthroughDF:                Yes
PassthroughDSCP:              Yes
***********************************************************************
TunnelID:                     M2
VpnActionName:                ManualTunnel-AH-SHA-AES~2
State:                        Active
HowToEncap:                   Tunnel
LocalEndPoint:                2001:db8:10::81:2:10
RemoteEndPoint:               2001:db8:10::81:8:10
HowToAuth:                    AH
 AuthAlgorithm:               HMAC-SHA1
 AuthInboundSpi:              67777      (0x   108C1)
 AuthOutboundSpi:             65555      (0x   10013)
HowToEncrypt:                 AES-CBC
 KeyLength:                   128
 EncryptInboundSpi:           66666      (0x   1046A)
 EncryptOutboundSpi:          64444      (0x    FBBC)
OutboundPackets:              0
OutboundBytes:                0
InboundPackets:               0
```

```
InboundBytes:              0
PassthroughDF:             n/a
PassthroughDSCP:           Yes
************************************************************************

2 entries selected
```

Manual tunnel (-m) primary option report field descriptions:
For more information about the header, see “The ipsec command report heading” on page 735.

TunnelID
The ID that uniquely defines the manual tunnel. In this example, TunnelID has the value `M` (for manual) followed by an arbitrary positive integer that was assigned when the manual tunnel was installed in the stack by the Policy Agent. A change to the manual tunnel policy definition results in a new tunnel ID.

VpnActionName
The name of the IpManVpnAction statement in POLICY that defines this manual tunnel.

State
This field has the value `Active` or `Inactive`. The value `Active` indicates that the tunnel is available for use between the local endpoint and remote endpoint. The value `Inactive` indicates that the tunnel is not available and must be activated.

LocalEndPoint
The local security endpoint address, as defined by the LocalSecurityEndpointAddr field of the IpManVpnAction statement.

RemoteEndPoint
The remote security endpoint address, as defined by the RemoteSecurityEndpointAddr field of the IpManVpnAction statement.

HowToEncap
Indicates the encapsulation mode for the tunnel. Possible values are `Transport` or `Tunnel`.

HowToAuth
Indicates the protocol headers that are used to carry authentication data. Possible values are `AH` or `ESP`.

AuthAlgorithm
Indicates the authentication algorithm that is being used. Possible values are:

- AES-XCBC-MAC-96
- HMAC-MD5
- HMAC-SHA1
- HMAC-SHA2-256-128
- HMAC-SHA2-384-192
- HMAC-SHA2-512-256

AuthInboundSpi
Indicates the local Security Parameter Index.

AuthOutboundSpi
Indicates the remote Security Parameter Index.

HowToEncrypt
Indicates whether encryption is to be used, and if so, the encryption algorithm that is used. Possible values are:

- n/a
- AES-CBC
- DES-CBC
- 3DES-CBC

EncryptInboundSpi
If encryption is used, this field indicates the local Security Parameter Index.

EncryptOutboundSpi
If encryption is used, this field indicates the remote Security Parameter Index.

KeyLength
The length, in bits, of the key used by the encryption algorithm. The length is specified as `n/a` for algorithms with a fixed key length.

OutboundPackets
The total number of outbound packets that have been protected by the tunnel.

OutboundBytes
The total number of outbound bytes that have been protected by the tunnel.

InboundPackets
The total number of inbound packets that have been protected by the tunnel.

InboundBytes
The total number of inbound bytes that have been protected by the tunnel.

PassthroughDF
Indicates whether the don't-fragment bit is copied from the inner IP header to the outer IP header in tunnel mode. Possible values are:

Yes The don't-fragment bit is copied to the outer IP header.

Clear The don't-fragment bit is cleared in the outer header.

Set The don't-fragment bit is set in the outer header.

n/a The tunnel is not IPv4 or is not in tunnel mode.

PassthroughDSCP
Indicates whether the differentiated services code point (DSCP) value is copied from the inner IP header to the outer IP header in tunnel mode. Possible values are:

Yes The DSCP value is copied to the outer IP header.

No The DSCP value is set to 0 in the outer IP header.

n/a The tunnel is not in tunnel mode.

IKE tunnel (-k) primary option

The **-k** primary option is used to display and manage IKE tunnels with respect to a particular TCP/IP stack. IKE tunnels are created to exchange key material on behalf of a dynamic tunnel. They are created using information from a

KeyExchangeRule statement, which is located based on the local and remote addresses and (if available) the local and remote IDs that satisfy the needs of a specific data request (for example, a dynamic tunnel need). IKE tunnels can be displayed, deactivated, and refreshed using their tunnel ID. They can also be displayed based on the KeyExchangeRule statement with which they are associated. IKE tunnels have only a representation in the IKE daemon and each is associated with a specific stack.

See “The z/OS UNIX ipsec command IKE tunnel (-k) option” on page 718 for parameter descriptions.

IKE tunnel (-k) primary option syntax:
For **-k** primary option syntax see “The z/OS UNIX ipsec command syntax” on page 700.

IKE tunnel (-k) primary option command examples:

ipsec -k display
: Displays the current IKE tunnels that are associated with the default stack.

ipsec -k display -z nsclient1 -n keyExRule2 -e
: Displays the IKE tunnel that is defined in policy by the KeyExchangeRule statement keyExRule2 value. Additionally, displays the dynamic tunnels that are associated with the IKE tunnel. The request is directed to the NSS server.

ipsec -k deactivate -a all
: Deactivates all IKE tunnels for the default stack. This causes all dynamic tunnels to be deactivated, effectively stopping all current dynamic tunnels and forcing new IKE tunnels to be created for any future activation.

ipsec -k refresh -p tcpcs1 -a K05
: Refresh the IKE tunnel for stack tcpcs1 that is identified by **K05** (the tunnel ID was obtained from an earlier **display** command). The current dynamic tunnels are not impacted by this request, but any new dynamic tunnel activation corresponding to K05 is associated with the refreshed IKE tunnel.

IKE tunnel (-k) primary option report example:

```
ipsec -p tcpcs4 -k display
```

```
CS V2R1 ipsec  Stack Name: TCPCS5  Fri Nov 25 17:40:56 2011
Primary:  IKE tunnel      Function: Display           Format:   Detail
Source:   IKED            Scope:    Current           TotAvail: n/a

TunnelID:                      K1
Generation:                    1
IKEVersion:                    1.0
KeyExchangeRuleName:           TunnelA~5
KeyExchangeActionName:         TunnelA
LocalEndPoint:                 10.83.5.3
LocalIDType:                   ID_IPV4_ADDR
LocalID:                       10.83.5.3
RemoteEndPoint:                10.83.4.3
RemoteIDType:                  ID_IPV4_ADDR
RemoteID:                      10.83.4.3
ExchangeMode:                  Aggressive
State:                         DONE
AuthenticationAlgorithm:       HMAC-SHA2-256-128
EncryptionAlgorithm:           AES-CBC
 KeyLength:                    128
PseudoRandomFunction:          HMAC-SHA2-256
```

```
DiffieHellmanGroup:           21
LocalAuthenticationMethod:    PresharedKey
RemoteAuthenticationMethod:   PresharedKey
InitiatorCookie:              0xD574468D783F7FB8
ResponderCookie:              0x24E4E9B5ABBB2F1C
Lifesize:                     0K
CurrentByteCount:             368b
Lifetime:                     480m
LifetimeRefresh:              2011/10/21 00:50:57
LifetimeExpires:              2011/10/21 01:11:47
ReauthInterval:               480m
ReauthTime:                   2011/10/21 00:50:57
Role:                         Initiator
AssociatedDynamicTunnels:     1
NATTSupportLevel:             RFC_zOS
NATInFrntLclScEndPnt:         No
NATInFrntRmtScEndPnt:         No
zOSCanInitiateP1SA:           Yes
AllowNat:                     Yes
RmtNAPTDetected:              No
RmtUdpEncapPort:              n/a
**********************************************************************
TunnelID:                     K15
Generation:                   1
IKEVersion:                   2.0
KeyExchangeRuleName:          TunnelB~5
KeyExchangeActionName:        TunnelB
LocalEndPoint:                10.84.5.2
LocalIDType:                  ID_FQDN
LocalID:                      b2.com
RemoteEndPoint:               10.185.4.20
RemoteIDType:                 ID_USER_FQDN
RemoteID:                     a3@a2.com
ExchangeMode:                 n/a
State:                        DONE
AuthenticationAlgorithm:      HMAC-MD5-96
EncryptionAlgorithm:          DES-CBC
 KeyLength:                   n/a
PseudoRandomFunction:         HMAC-MD5
DiffieHellmanGroup:           1
LocalAuthenticationMethod:    PresharedKey
RemoteAuthenticationMethod:   PresharedKey
InitiatorCookie:              0xE04F0DB48ADB8FF9
ResponderCookie:              0x33703BC726AB21EA
Lifesize:                     0K
CurrentByteCount:             325b
Lifetime:                     480m
LifetimeRefresh:              2011/10/21 01:28:50
LifetimeExpires:              2011/10/21 01:40:05
ReauthInterval:               0m
ReauthTime:                   n/a
Role:                         Responder
AssociatedDynamicTunnels:     1
NATTSupportLevel:             IKEv2
NATInFrntLclScEndPnt:         Yes
NATInFrntRmtScEndPnt:         Yes
zOSCanInitiateP1SA:           No
AllowNat:                     Yes
RmtNAPTDetected:              Yes
RmtUdpEncapPort:              1123
**********************************************************************

2 entries selected
```

IKE tunnel (-k) primary option report field descriptions:

For more information about the header, see “The ipsec command report heading” on page 735.

TunnelID
The ID that uniquely defines the IKE tunnel. In this example, TunnelID has the value `K` (for IKE) followed by an arbitrary positive integer that was assigned by the system when the tunnel was defined. This is the name to use when specifying an **ipsec** command selection criteria using the **-a** option.

Generation
This number is used to differentiate SAs for the same tunnel. The first SA that is created for a given tunnel is number 1.

IKEVersion
Specifies the IKE major and minor version that is used to negotiate the tunnel. Possible values are:

1.x IKE version 1

2.x IKE version 2

KeyExchangeRuleName
The name of the KeyExchangeRule statement that was used to define and control the characteristics of the IKE tunnel. The KeyExchangeRuleName value is established at the time the IKE tunnel is established.

KeyExchangeActionName
The name of the KeyExchangeAction statement that was used to initiate the IKE tunnel. The KeyExchangeActionName value is established at the time the IKE tunnel is established.

LocalEndpoint
The local security endpoint address of the IKE tunnel.

LocalIDType
Specifies the type of the local identity. Possible values are:

ID_IPV4_ADDR
An IPv4 address.

ID_IPV6_ADDR
An IPv6 address.

ID_FQDN
A fully qualified domain name.

ID_USER_FQDN
A user at a fully qualified domain name.

ID_DER_ASN1_DN
An X.500 distinguished name.

ID_KEY_ID
A vendor-specific value used to perform certain proprietary forms of identification.

LocalID
Specifies the value of the local identity.

Restriction: If the LocalIDType value is ID_KEY_ID, the LocalID value is truncated to avoid spanning multiple lines on a typical display device. An ellipsis is appended to this value to indicate that it was truncated. To display the entire value, use the wide (-r wide) display format.

RemoteEndpoint
The remote security endpoint address of the IKE tunnel.

RemoteIDType
Specifies the type of the remote identity. Possible values are:

ID_IPV4_ADDR
An IPv4 address.

ID_IPV6_ADDR
An IPv6 address.

ID_FQDN
A fully qualified domain name.

ID_USER_FQDN
A user at a fully qualified domain name.

ID_DER_ASN1_DN
An X.500 distinguished name.

ID_KEY_ID
A vendor-specific value used to perform certain proprietary forms of identification.

RemoteID
Specifies the value of the remote identity.

Restriction: If the RemoteIDType value is ID_KEY_ID, then the RemoteID value is truncated to avoid spanning multiple lines on a typical display device. An ellipsis is appended to this value to indicate that it was truncated. To display the entire value, use the wide (-r wide) display format.

ExchangeMode
The exchange mode used to negotiate the IKE tunnel. Possible values for an IKEv1 tunnel are Aggressive or Main. This field is supported for IKEv1 tunnels only and is always set to `n/a` for IKEv2 tunnels.

State The state of the tunnel with respect to the negotiation that occurs during activation.

Possible values for an IKEv1 tunnel are:

INIT Indicates that no key exchange messages have been initiated.

WAIT SA
Indicates that the first key exchange message has been sent and the endpoint is waiting for a response.

IN KE Indicates that a key exchange response has been sent.

WAIT KE
Indicates that a key exchange message has been sent and that the endpoint is waiting for a response.

DONE
Indicates that all key exchange messages have been completed and that the tunnel is available for data traffic.

EXPIRED
Indicates that tunnel has exceeded its lifetime and is not available for data traffic.

Possible values for an IKEv2 tunnel are:

INIT Indicates that no key exchange messages have been initiated.

WAIT KE
Indicates that an SA Init request is in progress.

WAIT AUTH
Indicates that an SA Auth request is in progress.

DONE
Indicates that all key exchange messages have been completed and that the tunnel is available for data traffic.

HALF-CLOSED
Indicates that the tunnel is in the process of closing.

EXPIRED
Indicates that tunnel has exceeded its lifetime and is not available for data traffic.

AuthenticationAlgorithm
Specifies the authentication algorithm that is used for authenticating IKE key exchange messages.

Possible values for IKEv1 tunnels are:

- HMAC-MD5
- HMAC-SHA1
- HMAC-SHA2-256-128
- HMAC-SHA2-384-192
- HMAC-SHA2-512-256

Possible values for IKEv2 tunnels are:

- AES128-XCBC-96
- HMAC-MD5-96
- HMAC-SHA1-96
- HMAC-SHA2-256-128
- HMAC-SHA2-384-192
- HMAC-SHA2-512-256

EncryptionAlgorithm
Specifies the encryption algorithm that is used for protecting IKE key exchange messages. Possible values are:

- `AES-CBC`
- `DES-CBC`
- `3DES-CBC`

KeyLength
The length, in bits, of the key used by the encryption algorithm. The length is specified as `n/a` for algorithms with a fixed key length.

PseudoRandomFunction
Specifies the pseudo-random function that is used for generating keying material. For IKEv1, the PseudoRandomFunction value is always the same value as the AuthenticationAlgorithm value. For IKEv2, the pseudo-random function is negotiated separately and might differ from the authentication algorithm. Possible values are:

- AES128-XCBC

- HMAC-MD5
- HMAC-SHA1
- HMAC-SHA2-256
- HMAC-SHA2-384
- HMAC-SHA2-512

DiffieHellmanGroup
Indicates the Diffie-Hellman group that is used during key exchange. If no Diffie-Hellman group is used, the value is 0.

LocalAuthenticationMethod
Indicates the method that the remote peer is using to authenticate the local endpoint. Possible values are

- PresharedKey
- RsaSignature
- ECDSA-256
- ECDSA-384
- ECDSA-521
- DigitalSignature

For IKEv1 tunnels, the authentication method is negotiated and it is always the same as the remote authentication method.

For IKEv2 tunnels, the authentication method is established by local policy and might differ from the remote authentication method.

RemoteAuthenticationMethod
Indicates the method that the local system is using to authenticate the remote endpoint. Possible values are:

- PresharedKey
- RsaSignature
- ECDSA-256
- ECDSA-384
- ECDSA-521
- Unknown - For IKEv2 tunnels only, the value Unknown is possible if the IKEv2 tunnel has not completed its initial exchanges.

For IKEv1 tunnels, the authentication method is negotiated and is always the same as the local authentication method.

For IKEv2 tunnels, the authentication method is established by policy on the remote peer and might differ from the local authentication method.

InitiatorCookie
During the phase 1 negotiation, the initiator created a cookie to identify itself during the exchange. This is the value of that cookie.

ResponderCookie
During the phase 1 negotiation, the responder created a cookie to identify itself during the exchange. This is the value of that cookie.

Lifesize
The number of kilobytes that can pass on the IKE tunnel before the tunnel must be refreshed. If the value is 0, then the refresh Lifesize value was `None` and byte counts are not used to monitor for tunnel refresh.

CurrentByteCount
The number of bytes that have been protected by the tunnel.

Lifetime
The number of minutes between each refresh.

LifetimeRefresh
The time at which the tunnel must be refreshed.

LifetimeExpires
The time at which the tunnel expires.

ReauthInterval
The number of minutes between each reauthentication.

ReauthTime
The time at which the tunnel must be reauthenticated.

Role Indicates whether this endpoint was the initiator or responder on the IKE tunnel negotiation.

AssociatedDynamicTunnels
A count of how many dynamic tunnels depend on this IKE tunnel for their maintenance.

NATTSupportLevel
The level of NAT traversal support agreed to during the phase 1 SA negotiation. The followling list shows the possible values:

D2RFC
Draft 2 of RFC 3947.

D3RFC
Draft 3 of RFC 3947.

RFC RFC 3947, with a non-z/OS remote security endpoint.

RFC_zOS
RFC 3947, with a z/OS remote security endpoint.

IKEv2 RFC 5996, with a non-z/OS remote security endpoint.

IKEv2_zOS
RFC 5996, with a z/OS remote security endpoint.

n/a NAT traversal is not supported for phase 1 SAs that use IPv6 addresses. This field has the value `n/a`.

None No NAT Traversal support.

NATInFrntLclScEndPnt
Indicates whether or not a NAT has been detected in front of the local security endpoint. NAT traversal is not supported for phase 1 SAs using IPv6 addresses. In this case, the field has the value `n/a`.

NATInFrntRmtScEndPnt
Indicates whether or not a NAT has been detected in front of the remote security endpoint. NAT traversal is not supported for phase 1 SAs using IPv6 addresses. In this case, the field has the value `n/a`.

zOSCanInitP1SA
Indicates whether z/OS can initiate the initial phase 1 SA negotiation. NAT traversal is not supported for phase 1 SAs that use IPv6 addresses. In this case, the field has the value `n/a`.

AllowNat
Indicates whether NAT traversal support is enabled. This field indicates the configured setting of the AllowNat keyword. NAT traversal is not supported for phase 1 SAs that use IPv6 addresses. In this case, the field has the value `n/a`.

RmtNAPTDetected
Indicates whether or not a NAT in front of the remote security endpoint has been detected performing port address translation. The value `Yes` indicates that port address translation by a NAT in front of the remote security endpoint NAT has been detected; the value `No` indicates that it has not been detected. NAT traversal is not supported for phase 1 SAs that use IPv6 addresses. In this case, the field has the value `n/a`.

RmtUdpEncapPort
The UDP-encapsulated port number used by the remote security endpoint. This field is valid only for NAT-traversal tunnels. Otherwise, this field has the value `n/a`.

Dynamic tunnel (-y) primary option

The **-y** primary option is used to display and manage dynamic tunnels with respect to a particular TCP/IP stack. Dynamic tunnels that have a LocalDynVpnRule name defined can be activated, deactivated, refreshed, and displayed by referencing that LocalDynVpnRule name. When the dynamic tunnel is active, it has a tunnel ID and that tunnel ID can be used as selection criteria for displaying, deactivating, and refreshing the dynamic tunnel. Dynamic tunnels can also be referenced by their associated IpDynVpnAction name for display purposes.

Dynamic tunnels have a representation in the stack and also in the IKE daemon, so there are two versions of the **display** command, which are controlled by the **-b** option. When **-b** is specified, the dynamic tunnel data is reported from the IKE daemon. Otherwise, the dynamic tunnel data is reported from the stack. For any dynamic tunnel, some of the report data is unique to the stack representation, some of the report data is unique to the IKE daemon representation, and some of the report data is common. In general, the common data from each representation of a specific tunnel should match. However, because of timing considerations, the common data for a specific tunnel might not be completely consistent between the stack report and the IKE daemon report.

See “The z/OS UNIX ipsec command dynamic tunnel (-y) option” on page 719 for parameter descriptions.

Dynamic tunnel (-y) primary option syntax:
For **-y** primary option syntax see “The z/OS UNIX ipsec command syntax” on page 700.

Dynamic tunnel (-y) primary option command examples:

ipsec -y display
Displays the current dynamic tunnel data from the default stack.

Tip: The **ipsec -y display -s** command displays the same information; the header would indicate `Display (shadows)`.

ipsec -y display -z nsclient1 -b
Displays the current dynamic tunnel data from the IKE daemon for NSS client nsclient1. The request is directed to the NSS server.

ipsec -y deactivate -a Y03

Deactivates the dynamic tunnel identified as `Y03` (the tunnel ID was obtained from an earlier **display** command) from the default stack.

ipsec -y activate -l localDynVpnRule2

Activates the dynamic tunnel that is defined by the LocalDynVpnRule statement named localDynVpnRule2 from the default stack.

ipsec -y refresh -l localDynVpnRule1 -z nsclient1

Refreshes the dynamic tunnel that is defined by the LocalDynVpnRule statement named localDynVpnRule1 for NSS client nsclient1. The request is directed to the NSS server.

Dynamic tunnel (-y) primary option report examples:

`ipsec -p tcpcs4 -y display`

```
CS V2R1 ipsec  Stack Name: TCPCS5  Fri Oct 21 17:41:00 2011
Primary:  Dynamic tunnel  Function: Display            Format:   Detail
Source:   Stack           Scope:    Current            TotAvail: 2

TunnelID:                     Y2
Generation:                   1
IKEVersion:                   1.0
ParentIKETunnelID:            K1
VpnActionName:                ESP-AES128-SHA2256
LocalDynVpnRule:              TunnelA
State:                        Active
HowToEncap:                   Transport
LocalEndPoint:                10.83.5.3
RemoteEndPoint:               10.83.4.3
LocalAddressBase:             10.83.5.3
LocalAddressPrefix:           n/a
LocalAddressRange:            n/a
RemoteAddressBase:            10.83.4.3
RemoteAddressPrefix:          n/a
RemoteAddressRange:           n/a
HowToAuth:                    ESP
 AuthAlgorithm:               HMAC-SHA2-256-128
 AuthInboundSpi:              71549037   (0x 443C06D)
 AuthOutboundSpi:             2153470104 (0x805B5898)
HowToEncrypt:                 AES-CBC
 KeyLength:                   128
 EncryptInboundSpi:           71549037   (0x 443C06D)
 EncryptOutboundSpi:          2153470104 (0x805B5898)
Protocol:                     ALL(0)
LocalPort:                    n/a
LocalPortRange:               n/a
RemotePort:                   n/a
RemotePortRange:              n/a
Type:                         n/a
TypeRange:                    n/a
Code:                         n/a
CodeRange:                    n/a
OutboundPackets:              1
OutboundBytes:                2008
InboundPackets:               1
InboundBytes:                 2008
Lifesize:                     0K
LifesizeRefresh:              0K
CurrentByteCount:             0b
LifetimeRefresh:              2011/10/20 20:44:48
LifetimeExpires:              2011/10/20 21:11:47
CurrentTime:                  2011/10/20 17:41:00
VPNLifeExpires:               2011/10/21 17:11:47
NAT Traversal Topology:
  UdpEncapMode:               No
  LclNATDetected:             No
  RmtNATDetected:             No
  RmtNAPTDetected:            No
  RmtIsGw:                    n/a
  RmtIsZOS:                   n/a
  zOSCanInitP2SA:             n/a
  RmtUdpEncapPort:            n/a
  SrcNATOARcvd:               n/a
  DstNATOARcvd:               n/a
PassthroughDF:                n/a
PassthroughDSCP:              n/a
************************************************************************
TunnelID:                     Y16
Generation:                   1
IKEVersion:                   2.0
ParentIKETunnelID:            K15
VpnActionName:                ESP-DES-MD5~133
LocalDynVpnRule:              n/a
State:                        Active
HowToEncap:                   Tunnel
LocalEndPoint:                10.84.5.2
RemoteEndPoint:               10.185.4.20
LocalAddressBase:             10.84.5.2
LocalAddressPrefix:           n/a
LocalAddressRange:            n/a
RemoteAddressBase:            10.185.4.20
RemoteAddressPrefix:          n/a
RemoteAddressRange:           n/a
HowToAuth:                    ESP
 AuthAlgorithm:               HMAC-MD5
 AuthInboundSpi:              1240535245 (0x49F110CD)
 AuthOutboundSpi:             2302713514 (0x89409EAA)
HowToEncrypt:                 DES-CBC
 KeyLength:                   n/a
 EncryptInboundSpi:           1240535245 (0x49F110CD)
 EncryptOutboundSpi:          2302713514 (0x89409EAA)
Protocol:                     ALL(0)
LocalPort:                    n/a
LocalPortRange:               n/a
RemotePort:                   n/a
RemotePortRange:              n/a
Type:                         n/a
TypeRange:                    n/a
Code:                         n/a
CodeRange:                    n/a
OutboundPackets:              4
OutboundBytes:                244
InboundPackets:               5
InboundBytes:                 300
Lifesize:                     0K
LifesizeRefresh:              0K
CurrentByteCount:             0b
LifetimeRefresh:              2011/10/20 21:15:24
LifetimeExpires:              2011/10/20 21:40:05
CurrentTime:                  2011/10/20 17:41:00
VPNLifeExpires:               2011/10/21 17:40:05
NAT Traversal Topology:
  UdpEncapMode:               Yes
  LclNATDetected:             Yes
  RmtNATDetected:             Yes
  RmtNAPTDetected:            Yes
  RmtIsGw:                    No
  RmtIsZOS:                   No
  zOSCanInitP2SA:             No
  RmtUdpEncapPort:            1123
  SrcNATOARcvd:               n/a
  DstNATOARcvd:              n/a
PassthroughDF:               Yes
PassthroughDSCP:             Yes
*********************************************************************

2 entries selected
```

Dynamic tunnel (-y) primary option report field descriptions:
For more information about the header, see "The ipsec command report heading" on page 735.

TunnelID
The ID that uniquely defines the dynamic tunnel. In this example, TunnelID has the value `Y` (for dynamic) followed by an arbitrary positive integer that was assigned by the system when the tunnel was defined. This is the name to use when specifying an **ipsec** command selection criteria using the **-a** option.

Generation
This number is used to differentiate SAs for the same tunnel. The first SA that is created for a given tunnel is number 1.

IKEVersion
Specifies the IKE major and minor version that is used to negotiate the tunnel. The possible values are:

1.x IKE version 1

2.x IKE version 2

ParentIKETunnelID
The tunnel ID of the phase 1 (IKE) tunnel that enables the creation of this dynamic tunnel.

VpnActionName
The name of the IpDynVpnAction statement in POLICY that defines this dynamic tunnel.

LocalDynVpnRule
The name of the LocalDynVpnRule statement with which the dynamic tunnel is associated. This is the name to use when specifying **ipsec** command selection criteria using the **-l** option. If the dynamic tunnel is not associated with a LocalDynVpnRule statement, the value is `n/a`. The LocalDynVpnRule value is established at the time the IKE tunnel is established.

State Possible state values are:

Active Indicates that the tunnel is available for use between the local endpoint and the remote endpoint.

Expired
Indicates that the tunnel reached its lifetime or lifesize value and could not be refreshed.

Refreshed
Indicates that the tunnel structure is not the current one representing the tunnel; another entry with the same TunnelID is considered current.

HowToEncap
Indicates the encapsulation mode for the tunnel. Possible values are `Transport` or `Tunnel`.

LocalEndPoint
The local security endpoint address of the dynamic tunnel.

RemoteEndPoint
The remote security endpoint address of the dynamic tunnel.

LocalAddressBase
The LocalAddressBase field describes the IP traffic protected by this dynamic tunnel. If the LocalAddressPrefix field and the LocalAddressRange field each display the value `n/a`, then the tunnel protects traffic with this single source address. Otherwise, the tunnel protects traffic with the source address described by the LocalAddressBase field and the corresponding mask or range.

LocalAddressPrefix
The LocalAddressBase field and the LocalAddressPrefix field describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic whose source address is in the range defined by the LocalAddressBase field and the subnet indicated by this prefix.

LocalAddressRange
The LocalAddressBase field and the LocalAddressRange field describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic in the range of IP addresses between the LocalAddressBase field value and this address (inclusive).

RemoteAddressBase
The RemoteAddressBase field describes the IP traffic protected by this dynamic tunnel. If the RemoteAddressPrefix field and RemoteAddressRange field display the value `n/a`, then the tunnel protects traffic with this single destination address. Otherwise, the tunnel protects traffic with the destination address described by the RemoteAddressBase field and the corresponding mask or range.

RemoteAddressPrefix
The RemoteAddressBase field and the RemoteAddressPrefix field describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic whose destination address is in the range defined by the RemoteAddressBase field and the subnet indicated by this prefix.

RemoteAddressRange
The RemoteAddressBase field and the RemoteAddressRange field describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic in the range of IP addresses between the RemoteAddressBase field value and this address (inclusive).

HowToAuth
Indicates what protocol headers are used to carry authentication data. Possible values are `AH` or `ESP`.

AuthAlgorithm
Indicates what authentication algorithm is being used. Possible values are:

- NULL
- AES-GMAC-128
- AES-GMAC-256

- AES-XCBC-MAC-96
- HMAC-MD5
- HMAC-SHA1
- HMAC-SHA2-256-128
- HMAC-SHA2-384-192
- HMAC-SHA2-512-256

AuthInboundSpi
Indicates the local Security Parameter Index.

AuthOutboundSpi
Indicates the remote Security Parameter Index.

HowToEncrypt
Indicates whether encryption is to be used, and if so, which encryption algorithm is used. Possible values are:

- AES-CBC
- AES-GCM-16
- DES-CBC
- 3DES-CBC
- NULL

  Note: This value is displayed when no encryption is used and ESP protocol headers are in use.
- `n/a`

  Note: This value is displayed when no encryption is used and AH protocol headers are in use.

EncryptInboundSpi
If encryption is being used, this field indicates the local Security Parameter Index.

EncryptOutboundSpi
If encryption is being used, this field indicates the remote Security Parameter Index.

KeyLength
The length, in bits, of the key used by the encryption algorithm. The length is specified as `n/a` for algorithms with a fixed key length, or if encryption is not used.

Protocol
Indicates the protocol that the dynamic tunnel is protecting. The value 0 indicates that the dynamic tunnel is protecting all protocols.

LocalPort
Indicates the source port number of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value `All` indicates that the dynamic tunnel is protecting all source ports. The value `Opaque` indicates that the dynamic tunnel is protecting only packets with unknown source ports caused by packet fragmentation. The value `n/a` indicates that this field is not applicable to the current protocol.

LocalPortRange
Indicates the upper value in the range of source port numbers of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or

UDP. The value n/a indicates that this field is not applicable to the current protocol or that the tunnel's source port selectors are entirely specified by the LocalPort value.

RemotePort
Indicates the destination port number of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value All indicates that the dynamic tunnel is protecting all destination ports. The value Opaque indicates that the dynamic tunnel is protecting only packets with unknown destination ports caused by packet fragmentation The value n/a indicates that this field is not applicable to the current protocol.

RemotePortRange
Indicates the upper value in the range of destination port numbers of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value n/a indicates that this field is not applicable to the current protocol or that the tunnel's destination port selectors are entirely specified by the RemotePort value.

Type Indicates the type of the data traffic that the dynamic tunnel is protecting when the protocol is ICMP, ICMPv6 or MIPv6. The value All indicates that the dynamic tunnel is protecting all types. The value Opaque indicates that the dynamic tunnel is protecting only packets with unknown types caused by packet fragmentation. The value n/a indicates that this field is not applicable to the current protocol.

TypeRange
Indicates the upper value in the type range of the data traffic that the dynamic tunnel is protecting when the Protocol value is ICMP, ICMPv6, or MIPv6. The value n/a indicates that this field is not applicable to the current protocol or that the type selectors for the tunnel are entirely specified by the Type value. This field is not applicable for protocols other than ICMP, ICMPv6, or MIPv6.

Code Indicates the specific type of the data traffic that the dynamic tunnel is protecting when the protocol is ICMP or ICMPv6. The value All indicates that the dynamic tunnel is protecting all codes. The value Opaque indicates that the dynamic tunnel is protecting only packets with unknown codes caused by packet fragmentation. The value n/a indicates that this field is not applicable to the current protocol.

CodeRange
Indicates the upper value in the range of code numbers of the data traffic that the dynamic tunnel is protecting when the protocol is ICMP or ICMPv6. The value n/a indicates that this field is not applicable to the current protocol, or else that the tunnel's type selectors are entirely specified by the Code value.

OutboundPackets
The total number of outbound packets that have been protected by the tunnel. This counter is maintained with the current tunnel structure so that when the tunnel is refreshed, the counter restarts at 0. The counter is updated only on the system at which the data traffic is encapsulated. In a SWSA environment, the field shows the total count for the local stack only (distributor or target), and not for any other distributed version of the tunnel in the sysplex.

OutboundBytes
The total number of outbound bytes that have been protected by the tunnel. This counter is maintained with the current tunnel structure so that

when the tunnel is refreshed, the counter restarts at 0. The counter is updated only on the system at which the data traffic is encapsulated. In a SWSA environment, the field shows the total count for the local stack only (distributor or target), and not for any other distributed version of the tunnel in the sysplex.

InboundPackets
The total number of inbound packets that have been protected by the tunnel. This counter is maintained with the current tunnel structure so that when the tunnel is refreshed, the counter restarts at 0. The counter is updated only on the system at which the data traffic is decapsulated. In a SWSA environment, the field shows the total count for the local stack only (distributor or target), and not for any other distributed version of the tunnel in the sysplex.

InboundBytes
The total number of inbound bytes that have been protected by the tunnel. This counter is maintained with the current tunnel structure so that when the tunnel is refreshed, the counter restarts at 0. The counter is updated only on the system at which the data traffic is decapsulated. In a SWSA environment, the field shows the total count for the local stack only (distributor or target), and not for any other distributed version of the tunnel in the sysplex.

Lifesize
The number of kilobytes that can be protected by the tunnel before it must be refreshed. If the value is 0, then the refresh Lifesize value was `None` and byte counts are not used to monitor for tunnel refresh.

LifesizeRefresh
The total number of kilobytes that can be protected by the tunnel before it is refreshed. This is the refresh Lifesize value minus a threshold that enables the tunnel refresh to occur without traffic disruption. If the value is 0, then the refresh Lifesize value was `None`; byte counts are not used to monitor for a tunnel refresh.

CurrentByteCount
The number of bytes that have been protected by the tunnel. If the Lifesize value is 0, then this value is also `0`.

LifetimeRefresh
A timestamp that indicates the time at which the tunnel must be refreshed. If the refresh time for the tunnel was 0, this value is `n/a`.

LifetimeExpires
A timestamp that indicates the time at which the tunnel expires. This is the LifetimeRefresh value minus a threshold that enables the tunnel refresh to occur without traffic disruption. If the refresh time for tunnel was 0, this value is `n/a`.

CurrentTime
A timestamp that indicates the current time of this display. This is for comparison with the values for the LifetimeRefresh, LifetimeExpires, and VPNLifeExpires fields.

VPNLifeExpires
A timestamp that indicates the time at which the tunnel expires and can no longer be used.

PassthroughDF
Indicates whether the don't-fragment bit is copied from the inner IP header to the outer IP header in tunnel mode. Possible values are:

Yes The don't-fragment bit is copied to the outer IP header.

Clear The don't-fragment bit is cleared in the outer header.

Set The don't-fragment bit is set in the outer header.

n/a The tunnel is not IPv4 or is not in tunnel mode.

PassthroughDSCP
Indicates whether the differentiated services code point (DSCP) value is copied from the inner IP header to the outer IP header in tunnel mode. Possible values are:

Yes The DSCP value is copied to the outer IP header.

No The DSCP value is set to 0 in the outer IP header.

n/a The tunnel is not in tunnel mode.

UdpEncapMode
Indicates whether or not UDP encapsulation is being applied to an SA to enable it to traverse a NAT.

LclNATDetected
Indicates whether or not a NAT has been detected in front of the local security endpoint.

RmtNATDetected
Indicates whether or not a NAT has been detected in front of the remote security endpoint.

RmtNAPTDetected
Indicates whether or not a NAT in front of the remote security endpoint has been detected performing port address translation. The value `Yes` indicates that port address translation by a NAT in front of the remote security endpoint NAT was detected; the value `No` indicates that it was not detected. A NAPT (network address protocol translator) can be detected by IKE, the stack, or by both. There might be cases where the stack detects NAPT but IKE does not detect NAPT, or where IKE detects NAPT but the stack does not detect NAPT. In these cases, the IKE NAPT settings might not match the stack's NAPT settings. However, this setting should be consistent within the IKE daemon for all tunnels negotiated with the same remote security endpoint IP address.

RmtIsGw
Indicates whether or not the remote security endpoint is acting as a security gateway when UDP encapsulation is being applied to an SA. If UDP encapsulation is not being used, this field displays the value `n/a`.

RmtIsZOS
Indicates whether or not the remote security endpoint is z/OS when UDP encapsulation is being applied to an SA. If UDP encapsulation is not being used, this field displays the value `n/a`.

zOSCanInitP2SA
Indicates whether or not z/OS can initiate the initial phase 2 SA negotiation.

RmtUdpEncapPort

The UDP-encapsulated port number used by the remote security endpoint. This field is valid only for NAT-traversal tunnels. Otherwise, this field displays the value `n/a`.

SrcNATOARcvd

For a UDP-encapsulated transport SA: Source NAT original IP address received during the IKE negotiation. The IKE peer sends the source IP address that it is aware of. If the IKE peer is behind a NAT device, this is the peer's private address. This value is 0.0.0.0 if a source NAT original IP address was not received. An IKE peer at a pre-RFC3947 NAT Traversal support level cannot send a source NAT-OA payload containing the original IP address. For information about accessing RFCs, see Appendix F, “Related protocol specifications,” on page 1073.

For a non-UDP-encapsulated transport SA: The value is `n/a`.

DstNATOARcvd

For a UDP-encapsulated transport SA: Destination NAT original IP address received during the IKE negotiation. The IKE peer sends the destination IP address that it is aware of. If this host is behind a NAT, the value displayed can be the public address of the host. This value is 0.0.0.0 if a destination NAT original IP address was not received. An IKE peer at a pre-RFC3947 NAT traversal support level cannot send a destination NAT-OA payload containing the original IP address. For information about accessing RFCs, see Appendix F, “Related protocol specifications,” on page 1073.

For a non-UDP-encapsulated transport SA: The value is `n/a`.

Dynamic tunnel (-y) primary option report examples:

`ipsec -p tcpcs -y display -b`

```
CS V2R1 ipsec  Stack Name: TCPCS5  Fri Oct 21 17:41:07 2011
Primary:  Dynamic tunnel  Function: Display          Format:   Detail
Source:   IKED            Scope:    Current          TotAvail: n/a

TunnelID:                    Y2
Generation:                  1
IKEVersion:                  1.0
ParentIKETunnelID:           K1
VpnActionName:               ESP-AES128-SHA2256
LocalDynVpnRule:             TunnelA
IpFilterRule:                TunnelA~8
State:                       DONE
HowActivated:                Command
HowToEncap:                  Transport
LocalEndPoint:               10.83.5.3
RemoteEndPoint:              10.83.4.3
LocalAddressBase:            10.83.5.3
LocalAddressPrefix:          n/a
LocalAddressRange:           n/a
RemoteAddressBase:           10.83.4.3
RemoteAddressPrefix:         n/a
RemoteAddressRange:          n/a
HowToAuth:                   ESP
 AuthAlgorithm:              HMAC-SHA2-256-128
 AuthInboundSpi:             71549037   (0x 443C06D)
 AuthOutboundSpi:            2153470104 (0x805B5898)
HowToEncrypt:                AES-CBC
 KeyLength:                  128
 EncryptInboundSpi:          71549037   (0x 443C06D)
 EncryptOutboundSpi:         2153470104 (0x805B5898)
```

```
Lifesize:                     0K
LifesizeRefresh:              0K
LifetimeRefresh:              2011/10/20 20:44:48
LifetimeExpires:              2011/10/20 21:11:47
CurrentTime:                  2011/10/20 17:41:07
VPNLifeExpires:               2011/10/21 17:11:47
AssociatedFiltProtocol:       ALL(0)
AssociatedFiltSrcPort:        n/a
AssociatedFiltSrcPortRange:   n/a
AssociatedFiltDestPort:       n/a
AssociatedFiltDestPortRange:  n/a
AssociatedFiltType:           n/a
AssociatedFiltTypeRange:      n/a
AssociatedFiltCode:           n/a
AssociatedFiltCodeRange:      n/a
PFS:                          No
DiffieHellmanGroup:           n/a
PendingNewActivation:         n/a
NAT Traversal Topology:
  UdpEncapMode:               No
  LclNATDetected:             No
  RmtNATDetected:             No
  RmtNAPTDetected:            No
  RmtIsGw:                    n/a
  RmtIsZOS:                   n/a
  zOSCanInitP2SA:             n/a
  RmtUdpEncapPort:            n/a
  SrcNATOARcvd:               n/a
  DstNATOARcvd:               n/a
  LclIpSpecExIDPayload:       n/a
  RmtIpSpecExIDPayload:       n/a
***********************************************************************
TunnelID:                     Y16
Generation:                   1
IKEVersion:                   2.0
ParentIKETunnelID:            K15
VpnActionName:                ESP-DES-MD5~133
LocalDynVpnRule:              n/a
IpFilterRule:                 TunnelB~8
State:                        DONE
HowActivated:                 Remote
HowToEncap:                   Tunnel
LocalEndPoint:                10.84.5.2
RemoteEndPoint:               10.185.4.20
LocalAddressBase:             10.84.5.2
LocalAddressPrefix:           n/a
LocalAddressRange:            n/a
RemoteAddressBase:            10.185.4.20
RemoteAddressPrefix:          n/a
RemoteAddressRange:           n/a
HowToAuth:                    ESP
 AuthAlgorithm:               HMAC-MD5
 AuthInboundSpi:              1240535245 (0x49F110CD)
 AuthOutboundSpi:             2302713514 (0x89409EAA)
HowToEncrypt:                 DES-CBC
 KeyLength:                   n/a
 EncryptInboundSpi:           1240535245 (0x49F110CD)
 EncryptOutboundSpi:          2302713514 (0x89409EAA)
Lifesize:                     0K
LifesizeRefresh:              0K
LifetimeRefresh:              2011/10/20 21:15:24
LifetimeExpires:              2011/10/20 21:40:05
CurrentTime:                  2011/10/20 17:41:07
VPNLifeExpires:               2011/10/21 17:40:05
AssociatedFiltProtocol:       ALL(0)
AssociatedFiltSrcPort:        n/a
AssociatedFiltSrcPortRange:   n/a
```

```
AssociatedFiltDestPort:       n/a
AssociatedFiltDestPortRange:  n/a
AssociatedFiltType:           n/a
AssociatedFiltTypeRange:      n/a
AssociatedFiltCode:           n/a
AssociatedFiltCodeRange:      n/a
PFS:                          Yes
DiffieHellmanGroup:           1
PendingNewActivation:         n/a
NAT Traversal Topology:
  UdpEncapMode:               Yes
  LclNATDetected:             Yes
  RmtNATDetected:             Yes
  RmtNAPTDetected:            Yes
  RmtIsGw:                    No
  RmtIsZOS:                   No
  zOSCanInitP2SA:             No
  RmtUdpEncapPort:            1123
  SrcNATOARcvd:               n/a
  DstNATOARcvd:               n/a
  LclIpSpecExIDPayload:       10.184.5.2
  RmtIpSpecExIDPayload:       10.85.4.23
**********************************************************************

2 entries selected
```

Dynamic tunnel (-y) primary option report field descriptions:
For more information about the header, see "The ipsec command report heading" on page 735.

TunnelID
The ID that uniquely defines the dynamic tunnel. In this example, the TunnelID has the value Y (for dynamic), followed by an arbitrary positive integer that was assigned by the system when the tunnel was defined. This is the name to use when specifying an **ipsec** command selection criteria using the **-a** option. The tunnel ID is Y0 unless the state is DONE.

Generation
This number is used to differentiate SAs for the same tunnel. The first SA that is created for a given tunnel is number 1.

IKEVersion
Specifies the IKE major and minor version that is used to negotiate the tunnel. The possible values are:

1.x IKE version 1

2.x IKE version 2

ParentIKETunnelID
The tunnel ID of the phase 1 (IKE) tunnel that enabled the creation of this dynamic tunnel.

VpnActionName
The name of the IpDynVpnAction statement in POLICY that defines this dynamic tunnel.

LocalDynVpnRule
The name of the LocalDynVpnRule statement with which the dynamic tunnel is associated. This is the name to use when specifying the **ipsec** command selection criteria using the **-l** option. If the dynamic tunnel is not associated with a LocalDynVpnRule statement, the value is n/a. The LocalDynVpnRule value is established at the time the IKE tunnel is established.

IpFilterRule
The name of the dynamic anchor rule in POLICY that controlled the creation of the dynamic tunnel. The IpFilterRule value is established at the time the IKE tunnel is established.

State This is the state of the tunnel with respect to the negotiation that occurs during activation. Possible values are:

INIT Indicates that no key exchange messages have been initiated.

KEP Indicates that key exchange messages are being processed, but that the full exchange has not completed.

DONE
Indicates that all key exchange messages have been completed and that the tunnel is usable for data traffic.

NOTIFY
Indicates that key exchange messages have been completed, but that until a connection notification is received from the tunnel endpoint, the tunnel is not done.

PENDING
Indicates that the report is for a dynamic tunnel request that is pending the activation of an IKE tunnel to allow it to begin; PENDING is the value only when the dynamic tunnel report is part of an IKE report that cascades the associated dynamic tunnels, where a pending dynamic tunnel request is shown with this state value.

HowActivated
Indicates the way in which this tunnel was activated. Possible values are:

Command
Indicates that the tunnel was activated as a result of an **ipsec** command invocation.

OnDemand
Indicates that the tunnel was activated to satisfy locally initiated data traffic.

Auto Indicates that the tunnel was activated automatically when the IKE daemon received configuration information from Policy Agent.

VIPA Indicates that the tunnel was activated as part of a SWSA takeover.

Remote
Indicates that the tunnel was initiated remotely and that this security endpoint was the responder in the negotiations.

HowToEncap
Indicates the encapsulation mode for the tunnel. Possible values are `Transport` or `Tunnel`.

LocalEndPoint
The local security endpoint address of the dynamic tunnel.

RemoteEndPoint
The remote security endpoint address of the dynamic tunnel.

LocalAddressBase
LocalAddressBase describes the IP traffic protected by this dynamic tunnel. If the LocalAddressPrefix and LocalAddressRange fields display the value `n/a`, then the tunnel protects traffic with this single source address.

Otherwise, the tunnel protects traffic with the source address described by the LocalAddressBase field and the corresponding mask or range.

LocalAddressPrefix
The LocalAddressBase and LocalAddressPrefix fields describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic whose source address is in the range defined by the LocalAddressBase field and the subnet indicated by this prefix.

LocalAddressRange
The LocalAddressBase and LocalAddressRange fields describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic in the range of IP addresses between the LocalAddressBase field value and this address (inclusive).

RemoteAddressBase
RemoteAddressBase describes the IP traffic protected by this dynamic tunnel. If the RemoteAddressPrefix and RemoteAddressRange fields display the value `n/a`, then the tunnel protects traffic with this single destination address. Otherwise, the tunnel protects traffic with the destination address described by the RemoteAddressBase field and the corresponding mask or range.

RemoteAddressPrefix
The RemoteAddressBase and RemoteAddressPrefix fields describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic whose destination address is in the range defined by the RemoteAddressBase field and the subnet indicated by this prefix.

RemoteAddressRange
The RemoteAddressBase and RemoteAddressRange fields describe the IP traffic protected by this dynamic tunnel. If this field does not display the value `n/a`, then the tunnel protects traffic in the range of IP addresses between the RemoteAddressBase field value and this address (inclusive).

HowToAuth
Indicates what protocol headers are used to carry authentication data. Possible values are `AH` or `ESP`.

AuthAlgorithm
Indicates what authentication algorithm is being used. Possible values are:

- NULL
- AES-GMAC-128
- AES-GMAC-256
- AES-XCBC-MAC-96
- HMAC-MD5
- HMAC-SHA1
- HMAC-SHA2-256-128
- HMAC-SHA2-384-192
- HMAC-SHA2-512-256

AuthInboundSpi
Indicates the local Security Parameter Index.

AuthOutboundSpi
Indicates the remote Security Parameter Index.

HowToEncrypt
Indicates whether encryption is to be used, and if so, which encryption algorithm is used. Possible values are:

- AES-CBC
- AES-GCM-16
- DES-CBC
- 3DES-CBC
- NULL

  Note: This value is displayed when no encryption is used and ESP protocol headers are in use.
- n/a

  Note: This value is displayed when no encryption is used and AH protocol headers are in use.

EncryptInboundSpi
If encryption is being used, this field indicates the remote Security Parameter Index.

EncryptOutboundSpi
If encryption is being used, this field indicates the local Security Parameter Index.

KeyLength
The length, in bits, of the key used by the encryption algorithm. The length is specified as `n/a` for algorithms with a fixed key length.

Lifesize
The number of kilobytes that can be protected by the tunnel before it must be refreshed. If the value is `0`, then the refresh Lifesize value was `None`; byte counts are not used to monitor for tunnel refresh.

LifesizeRefresh
The total number of kilobytes that can be protected by the tunnel before it is refreshed. This is the refresh Lifesize value minus a threshold that enables the tunnel refresh to occur without traffic disruption. If the value is `0`, then the negotiated refresh Lifesize value was `None`; byte counts are not used to monitor for tunnel refresh.

LifetimeRefresh
A timestamp indicating the time at which the tunnel must be refreshed. This is the lifetime expire value minus a threshold that enables the tunnel refresh to occur without traffic disruption.

LifetimeExpires
A timestamp indicating the time at which the tunnel expires.

CurrentTime
A timestamp indicating the current time of this display. This is for comparison with LifetimeRefresh, LifetimeExpires, and VPNLifeExpires values.

VPNLifeExpires
A timestamp indicating the time at which the tunnel expires and can no longer be used.

AssociatedFiltProtocol
Indicates the protocol that is being protected by the dynamic tunnel. The value `0` indicates that the dynamic tunnel is protecting all protocols.

AssociatedFiltSrcPort
Indicates the source port number of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value `All` indicates that the dynamic tunnel is protecting all source ports. The value `Opaque` indicates that the dynamic tunnel is protecting only packets with unknown source ports caused by packet fragmentation. The value `n/a` indicates that this field is not applicable to the current protocol.

AssociatedFiltSrcPortRange
Indicates the upper value in the range of source port numbers of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value `n/a` indicates that this field is not applicable to the current protocol, or that the tunnel's source port selectors are entirely specified by the AssociatedFiltSrcPort value.

AssociatedFiltDestPort
Indicates the destination port number of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value `All` indicates that the dynamic tunnel is protecting all destination ports. The value `Opaque` indicates that the dynamic tunnel is protecting only packets with unknown destination ports caused by packet fragmentation. The value `n/a` indicates that this field is not applicable to the current protocol.

AssociatedFiltDestPortRange
Indicates the upper value in the range of destination port numbers of the data traffic that the dynamic tunnel is protecting when the protocol is TCP or UDP. The value `n/a` indicates that this field is not applicable to the current protocol or that the tunnel's destination port selectors are entirely specified by the AssociatedFiltDestPort value.

AssociatedFiltType
Indicates the type of the data traffic that the dynamic tunnel is protecting when Protocol is ICMP, ICMPv6, or MIPv6. The value `ALL` indicates that the dynamic tunnel is protecting all types. The value `Opaque` indicates that the dynamic tunnel is protecting only packets with unknown source ports caused by packet fragmentation. This field is not applicable for protocols other than ICMP and ICMPv6..

AssociatedFiltTypeRange
Indicates the upper value in the range of type numbers of the data traffic that the dynamic tunnel is protecting when the protocol is ICMP, ICMPv6, or MIPv6. The value `n/a` indicates that this field is not applicable to the current protocol, or that the tunnel's type selectors are entirely specified by the AssociatedFiltType value.

AssociatedFiltCode
Indicates the code of the data traffic that the dynamic tunnel is protecting when Protocol is ICMP or ICMPv6. The value `ALL` indicates that the dynamic tunnel is protecting all codes. The value `Opaque` indicates that the dynamic tunnel is protecting only packets with an unknown code caused by packet fragmentation. This field is not applicable for protocols other than ICMP, ICMPv6, or MIPv6.

AssociatedFiltCodeRange
Indicates the upper value in the range of code numbers of the data traffic that the dynamic tunnel is protecting when the protocol is ICMP or ICMPv6. The value n/a indicates that this field is not applicable to the current protocol, or that the tunnel's code selectors are entirely specified by the AssociatedFiltCode value.

PFS Indicates whether the dynamic tunnel is using perfect forward secrecy. If the key exchange methodology uses a Diffie-Hellman group, then the PFS value is Yes.

DiffieHellmanGroup
Indicates the DiffieHellmanGroup that is used during key exchange. If no group is being used, the value is 0.

PendingNewActivation
Indicates whether the phase 2 is for a new activation attempt. The value Yes indicates that the phase 2 is a new activation attempt; the value No indicates that it is not a new activation attempt. This field is valid only when the State value is PENDING. Otherwise, this field has the value n/a.

UdpEncapMode
Indicates whether or not UDP encapsulation is being applied to an SA to enable it to traverse a NAT. NAT traversal is not supported for phase 2 SAs using IPv6 addresses. In this case, the field has the value n/a.

LclNATDetected
Indicates whether or not a NAT has been detected in front of the local security endpoint. NAT traversal is not supported for phase 2 SAs using IPv6 addresses. In this case, the field has the value n/a.

RmtNATDetected
Indicates whether or not a NAT has been detected in front of the remote security endpoint. NAT traversal is not supported for phase 2 SAs using IPv6 addresses. In this case, the field has the value n/a.

RmtNAPTDetected
Indicates whether or not a NAT in front of the remote security endpoint has been detected performing port address translation. A value of Yes indicates that port address translation by a NAT in front of the remote security endpoint NAT was detected; the value No indicates that it was not detected. A NAPT (network address protocol translator) can be detected by IKE, the stack, or by both. There might be cases where the stack detects NAPT but IKE does not, or where IKE detects NAPT but the stack does not. In these cases, the IKE NAPT settings might not match the stack's NAPT settings. However, this setting should be consistent within the IKE daemon for all tunnels negotiated with the same remote security endpoint IP address.NAT traversal is not supported for phase 2 SAs using IPv6 addresses. In this case, the field has the value n/a.

RmtIsGw
Indicates whether or not the remote security endpoint is acting as a security gateway when UDP encapsulation is being applied to an SA. If UDP encapsulation is not being used, this field displays the value n/a.

RmtIsZOS
Indicates whether or not the remote security endpoint is z/OS when UDP encapsulation is being applied to an SA. If UDP encapsulation is not being used, this field displays the value n/a.

zOSCanInitP2SA

Indicates whether or not z/OS can initiate the initial phase 2 SA negotiation. If UDP encapsulation is not being used, this field displays the value `n/a`.

RmtUdpEncapPort

The UDP-encapsulated port number used by the remote security endpoint. If UDP encapsulation is not being used, this field displays the value `n/a`.

SrcNATOARcvd

For a UDP-encapsulated transport SA: Source NAT original IP address received during the IKE negotiation. The IKE peer sends the source IP address that it is aware of. If the IKE peer is behind a NAT device, this is the peer's private address. This value is 0.0.0.0 if a source NAT original IP address was not received. An IKE peer at a pre-RFC3947 NAT Traversal support level cannot send a source NAT-OA payload containing the original IP address. For information about accessing RFCs, see Appendix F, “Related protocol specifications,” on page 1073.

For a non-UDP-encapsulated transport SA: The value is `n/a`.

DstNATOARcvd

For a UDP-encapsulated transport SA: Destination NAT original IP address received during the IKE negotiation. The IKE peer sends the destination IP address that it is aware of. If this host is behind a NAT, the value displayed can be the public address of the host. This value is 0.0.0.0 if a destination NAT original IP address was not received. An IKE peer at a pre-RFC3947 NAT traversal support level cannot send a destination NAT-OA payload containing the original IP address. For information about accessing RFCs, see Appendix F, “Related protocol specifications,” on page 1073.

For a non-UDP-encapsulated transport SA: The value is `n/a`.

LclIpSpecExIDPayload

The local IP specification exchanged by the peers. If the SA is IKEv1, and if no ID payloads were exchanged, this field displays the value 0. The local IP specification is always a single IP address. If UDP encapsulation is not being used, this field displays the value `n/a`.

RmtIpSpecExIDPayload

The remote IP specification exchanged by the peers. If the SA is IKEv1, and if no ID payloads were exchanged, this field displays the value 0. The remote IP specification can be a single IP address, IP address range, IP address followed by a slash and number of bits mask, host name, RFC821 name, or distinguished name. If UDP encapsulation is not being used, this field displays the value `n/a`. For information about accessing RFCs, see Appendix F, “Related protocol specifications,” on page 1073.

Interface (-i) primary option

The **-i** primary option is used to display interface information that is defined to the specified TCP/IP stack. Interface configuration is obtained from the TCPIP profile. This interface display applies only to stacks that are configured with IPSECURITY.

See “The z/OS UNIX ipsec command interface (-i) option” on page 721 for parameter descriptions.

Interface (-i) primary option syntax:

For **-i** primary option syntax see “The z/OS UNIX ipsec command syntax” on page 700.

Interface (-i) primary option command examples:

ipsec -i display

Displays the interface definition data from the default stack.

ipsec -i display -z nsclient1

Displays the interface definition data for the NSS client nsclient1. The request is directed to the NSS server.

Interface (-i) primary option report example:

```
ipsec -p tcpcs4 -i display
```

```
CS V2R1 ipsec  Stack Name: TCPCS4  Fri Nov 25 07:13:15 2011
Primary:  Interface       Function: Display            Format:   Detail
Source:   Stack           Scope:    Current            TotAvail: 5

***********************************************************************
InterfaceName                 MPC4124L
SecurityClass                 255
Active                        Yes
DVIPA                         No
Address                       10.11.2.4
***********************************************************************
InterfaceName                 TOVTAM
SecurityClass                 255
Active                        No
DVIPA                         No
Address                       10.51.0.4
***********************************************************************
InterfaceName                 IUTSAMEH6
SecurityClass                 255
Active                        No
DVIPA                         No
Address                       2001:db8:10::51:0:4
***********************************************************************
InterfaceName                 MPC6124
SecurityClass                 255
Active                        Yes
DVIPA                         No
Address                       2001:db8:10::11:2:4
***********************************************************************
InterfaceName                 MPC6124
SecurityClass                 255
Active                        Yes
DVIPA                         No
Address                       fe80::11:2:4
***********************************************************************

5 entries selected
```

Interface (-i) primary option report field descriptions:
For more information about the header, see "The ipsec command report heading" on page 735.

InterfaceName

The name of the interface as defined on the system. The name is from a LINK or INTERFACE statement that corresponds to a HOME address in the profile.

SecurityClass

The value is in the range 1 - 255. Traffic over the interface matches a filter rule with the same security class value as the interface or a filter rule with a security class value 0.

- For IPv4, the security class is defined on the LINK statement or the IPCONFIG statement (with DYNAMICXCF)
- For IPv6, the security class is defined on the INTERFACE statement or on the IPCONFIG6 statement (with DYNAMICXCF).

Active Indicates whether the interface is active or not.

DVIPA
Indicates whether the Address field represents a dynamic virtual IP address.

Address
The IP address of the interface.

IP traffic test (-t) primary option

The **-t** primary option is used to indirectly query the current filter rules to determine whether a rule exists that applies to a particular kind of data traffic. Given a source and destination address, a protocol, and (if the protocol requires it) a source and destination port pair, all of the filter rules that apply to that kind of data traffic are displayed in the order in which they would be applied. The search can be further qualified by specifying whether the traffic is outbound or inbound by security class.

See “The z/OS UNIX ipsec command IP traffic test (-t) option” on page 721 for parameter descriptions.

IP traffic test (-t) primary option syntax:
For **-t** primary option syntax see “The z/OS UNIX ipsec command syntax” on page 700.

IP traffic test (-t) primary option command examples:

ipsec -t 10.0.0.1 10.0.0.2 icmp
Displays the current filters that apply to ICMP traffic between two addresses from the default stack.

ipsec -t 10.0.0.1 10.0.0.2 tcp 1024 1025 -z nsclient1
Displays the current filters that apply to TCP traffic on the specified ports between two addresses from the IP stack for the NSS client nsclient1. The request is directed to the NSS server.

ipsec -t 2001::1:1 2001::1:2 udp 1026 1027
Displays the current filters that apply to UDP traffic on the specified ports between two IPv6 addresses from the default stack.

IP traffic test (-t) primary option report examples:

```
ipsec -p tcpcs4 -t 2001:db8:10::81:2:6 2001:db8:10::81:8:6 tcp 1027 21 out
```

```
CS V2R1 ipsec  Stack Name: TCPCS4  Tue Feb 14 07:13:50 2012
Primary:  IP Traffic Test Function: Display            Format:   Detail
Source:   Stack Policy    Scope:    n/a                TotAvail: 5
TestData: 2001:db8:10::81:2:6  2001:db8:10::81:8:6  tcp 1027 21 out
Defensive Mode: Inactive

FilterName:                   IPSecGWv6~7
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv6~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y3
Type:                         Dynamic
```

```
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          n/a
SourceAddress:                2001:db8:10::81:2:0
SourceAddressPrefix:          112
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  2001:db8:10::81:8:1
DestAddressPrefix:            n/a
DestAddressRange:             2001:db8:10::81:8:6
DestAddressGranularity:       n/a
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   IPSecGWv6~7
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv6~6
VpnActionName:                ESP-Hmac_Md5-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
```

```
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                2001:db8:10::81:2:0
SourceAddressPrefix:          112
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  2001:db8:10::81:8:1
DestAddressPrefix:            n/a
DestAddressRange:             2001:db8:10::81:8:6
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
************************************************************************
FilterName:                   IP-Range2all6~6
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IP-Range2all6~5
VpnActionName:                ESP-Hmac_Sha-3DES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                2001:db8:10::81:2:1
SourceAddressPrefix:          n/a
SourceAddressRange:           2001:db8:10::81:2:6
SourceAddressGranularity:     Rule
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
```

```
DestAddress:                  ::
DestAddressPrefix:            0
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   all2Subnet6~6
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         all2Subnet6~5
VpnActionName:                ESP-Hmac_Sha-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          Rule
ICMPCode:                     n/a
ICMPCodeGranularity:          Rule
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Packet
SourceAddress:                ::
SourceAddressPrefix:          0
SourceAddressRange:           n/a
SourceAddressGranularity:     Rule
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  2001:db8:10::81:8:0
DestAddressPrefix:            112
DestAddressRange:             n/a
DestAddressGranularity:       Rule
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
```

```
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         Rule
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   DenyAllRule_Generated__________Outbnd_v6
FilterNameExtension:          n/a
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                n/a
TunnelID:                     0x00
Type:                         Generic
DefensiveType:                n/a
State:                        Active
Action:                       Deny
Scope:                        Both
Direction:                    Outbound
OnDemand:                     n/a
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                ::
SourceAddressPrefix:          0
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  ::
DestAddressPrefix:            0
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 18:13:12
UpdateTime:                   2012/02/14 18:13:12
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
```

```
AssociatedStackCount:        n/a
************************************************************************

5 entries selected
```

```
ipsec -p tcpcs -t 9.27.153.14 9.42.105.78  tcp 1035 23 in 0
```

```
CS V2R1 ipsec  Stack Name: TCPCS  Tue Feb 14 15:26:53 2012
Primary:  IP Traffic Test Function: Display           Format:   Detail
Source:   Stack Policy    Scope:    n/a               TotAvail: 6
TestData: 9.27.153.14  9.42.105.78  tcp 1035 23 in 0
Defensive Mode: Inactive

FilterName:                  odessa-ipsec
FilterNameExtension:         2
GroupName:                   n/a
LocalStartActionName:        n/a
VpnActionName:               IPSec__Gold
TunnelID:                    Y2
Type:                        NATT Dynamic
DefensiveType:               n/a
State:                       Active
Action:                      Permit
Scope:                       Local
Direction:                   Inbound
OnDemand:                    Yes
SecurityClass:               0
Logging:                     All
LogLimit:                    n/a
Protocol:                    All
ICMPType:                    n/a
ICMPTypeGranularity:         n/a
ICMPCode:                    n/a
ICMPCodeGranularity:         n/a
OSPFType:                    n/a
TCPQualifier:                n/a
ProtocolGranularity:         n/a
SourceAddress:               9.27.153.14
SourceAddressPrefix:         n/a
SourceAddressRange:          n/a
SourceAddressGranularity:    n/a
SourcePort:                  n/a
SourcePortRange:             n/a
SourcePortGranularity:       n/a
DestAddress:                 9.42.105.78
DestAddressPrefix:           n/a
DestAddressRange:            n/a
DestAddressGranularity:      n/a
DestPort:                    n/a
DestPortRange:               n/a
DestPortGranularity:         n/a
OrigRmtConnPort:             n/a
RmtIDPayload:                10.37.55.211
RmtUdpEncapPort:             4500
CreateTime:                  n/a
UpdateTime:                  n/a
DiscardAction:               Silent
MIPv6Type:                   n/a
MIPv6TypeGranularity:        n/a
TypeRange:                   n/a
CodeRange:                   n/a
RemoteIdentityType:          n/a
RemoteIdentity:              n/a
FragmentsOnly:               No
FilterMatches:               0
LifetimeExpires:             n/a
AssociatedStackCount:        n/a
************************************************************************
```

```
FilterName:                   odessa-ipsec
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y4
Type:                         NATT Dynamic
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Inbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                9.27.153.14
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.42.105.78
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 10.37.55.212
RmtUdpEncapPort:              4500
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y0
Type:                         NATT Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Inbound
```

```
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                9.27.153.14
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  9.42.105.78
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   n/a
UpdateTime:                   n/a
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                7
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   odessa-ipsec
FilterNameExtension:          2
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                IPSec__Gold
TunnelID:                     Y0
Type:                         Dynamic Anchor
DefensiveType:                n/a
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Inbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
```

```
SourceAddress:               9.27.153.14
SourceAddressPrefix:         n/a
SourceAddressRange:          n/a
SourceAddressGranularity:    Packet
SourcePort:                  n/a
SourcePortRange:             n/a
SourcePortGranularity:       n/a
DestAddress:                 9.42.105.78
DestAddressPrefix:           n/a
DestAddressRange:            n/a
DestAddressGranularity:      Packet
DestPort:                    n/a
DestPortRange:               n/a
DestPortGranularity:         n/a
OrigRmtConnPort:             n/a
RmtIDPayload:                n/a
RmtUdpEncapPort:             n/a
CreateTime:                  2012/02/14 15:17:06
UpdateTime:                  2012/02/14 15:17:06
DiscardAction:               Silent
MIPv6Type:                   n/a
MIPv6TypeGranularity:        n/a
TypeRange:                   n/a
CodeRange:                   n/a
RemoteIdentityType:          n/a
RemoteIdentity:              n/a
FragmentsOnly:               No
FilterMatches:               7
LifetimeExpires:             n/a
AssociatedStackCount:        n/a
************************************************************************
FilterName:                  all4
FilterNameExtension:         2
GroupName:                   n/a
LocalStartActionName:        n/a
VpnActionName:               n/a
TunnelID:                    0x00
Type:                        Generic
DefensiveType:               n/a
State:                       Active
Action:                      Permit
Scope:                       Local
Direction:                   Inbound
OnDemand:                    n/a
SecurityClass:               0
Logging:                     None
LogLimit:                    n/a
Protocol:                    All
ICMPType:                    n/a
ICMPTypeGranularity:         n/a
ICMPCode:                    n/a
ICMPCodeGranularity:         n/a
OSPFType:                    n/a
TCPQualifier:                n/a
ProtocolGranularity:         n/a
SourceAddress:               0.0.0.0
SourceAddressPrefix:         0
SourceAddressRange:          n/a
SourceAddressGranularity:    n/a
SourcePort:                  n/a
SourcePortRange:             n/a
SourcePortGranularity:       n/a
DestAddress:                 0.0.0.0
DestAddressPrefix:           0
DestAddressRange:            n/a
DestAddressGranularity:      n/a
DestPort:                    n/a
```

```
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 15:17:06
UpdateTime:                   2012/02/14 15:17:06
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                11
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************
FilterName:                   DenyAllRule_Generated___________Inbnd
FilterNameExtension:          n/a
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                n/a
TunnelID:                     0x00
Type:                         Generic
DefensiveType:                n/a
State:                        Active
Action:                       Deny
Scope:                        Both
Direction:                    Inbound
OnDemand:                     n/a
SecurityClass:                0
Logging:                      All
LogLimit:                     n/a
Protocol:                     All
ICMPType:                     n/a
ICMPTypeGranularity:          n/a
ICMPCode:                     n/a
ICMPCodeGranularity:          n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          n/a
SourceAddress:                0.0.0.0
SourceAddressPrefix:          0
SourceAddressRange:           n/a
SourceAddressGranularity:     n/a
SourcePort:                   n/a
SourcePortRange:              n/a
SourcePortGranularity:        n/a
DestAddress:                  0.0.0.0
DestAddressPrefix:            0
DestAddressRange:             n/a
DestAddressGranularity:       n/a
DestPort:                     n/a
DestPortRange:                n/a
DestPortGranularity:          n/a
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2012/02/14 15:17:06
UpdateTime:                   2012/02/14 15:17:06
DiscardAction:                Silent
MIPv6Type:                    n/a
MIPv6TypeGranularity:         n/a
TypeRange:                    n/a
CodeRange:                    n/a
```

```
RemoteIdentityType:           n/a
RemoteIdentity:               n/a
FragmentsOnly:                No
FilterMatches:                0
LifetimeExpires:              n/a
AssociatedStackCount:         n/a
***********************************************************************

6 entries selected
```

IP traffic test (-t) primary option report field descriptions:
For a traffic test display, the third heading line of the report shows the command request options that were used to make the filters search. For the rest of the header information, see "The ipsec command report heading" on page 735.

The results of a traffic test display is the set of filters that apply to the input test data. The filters are shown in the order in which they are applied by the stack. See "IP filter (-f) primary option" on page 738 for a description of the fields in each filter.

NATT port translation (-o) primary option

The **-o** primary option is used to display the selected NAT traversal remote port translations. If no remote IP address is specified (using the **-q** option), all NAT traversal remote port translations are displayed. If there is a selected remote IP address (using the **-q** option), or a selected remote IP address with one or more ports (using the **-q -u** options), then the selected NAT traversal remote port translation information is displayed.

See "The z/OS UNIX ipsec command NATT port translation (-o) option" on page 722 for parameter descriptions.

NATT port translation (-o) primary option syntax:
For **-o** primary option syntax see "The z/OS UNIX ipsec command syntax" on page 700.

NATT port translation (-o) primary option command syntax examples:

ipsec -o display
Displays NAT traversal remote port translations from the default stack.

ipsec -o display -z nsclient1 -q 1.1.1.1
Displays NAT traversal remote port translations for the specified IP address from the NSS client nsclient1. The request is directed to the NSS server.

ipsec -o display -q 1.1.1.1 -u 202 203
Displays NAT traversal remote port translations for the specified IP address and ports from the default stack.

NATT port translation (-o) primary option report examples:

```
CS V2R1 ipsec  Stack Name: TCPCS5  Fri Nov 25 17:40:14 2011
Primary:  NATT Port Trans Function: Display            Format:   Detail
Source:   Stack           Scope:    Current            TotAvail: 3

RmtIpAddress:         10.185.4.20
Protocol:             TCP(6)
TransRmtConnPort:     5130
OrigRmtConnPort:      5130
RmtInnerIpAddress:    10.85.4.21
***********************************************************************
RmtIpAddress:         10.185.4.20
```

```
Protocol:            TCP(6)
TransRmtConnPort:    5330
OrigRmtConnPort:     5330
RmtInnerIpAddress:   10.85.4.23
*****************************************************************
RmtIpAddress:        10.185.4.20
Protocol:            TCP(6)
TransRmtConnPort:    65535
OrigRmtConnPort:     5130
RmtInnerIpAddress:   10.85.4.22
*****************************************************************

3 entries selected
```

NATT port translation (-o) primary option report field descriptions:
For the header information, see "The ipsec command report heading" on page 735.

RmtIpAddress
The public IP address assigned by NAT.

Protocol
TCP or UDP from the inner IP header of an inbound packet.

TransRmtConnPort
The translated port assigned by NAT traversal port translation processing. If different than the OrigRmtConnPort value, another client from the same remote public IP address was already using the original remote port. A translated remote port is assigned rather than rejecting the second client's request.

OrigRmtConnPort
The peer's original connection remote port.

RmtInnerIpAddress

- For tunnel mode, the remote IP address from the inner IP header of an inbound packet.
- For transport mode, the private address of the peer that is from the source NAT-OA payload received during the IKEv1 negotiation or from the Traffic Selector payload received during the IKEv2 negotiation. If the peer initiated the phase 2 IKEv2 negotiation, the address is from the TSi payload. Otherwise, if the local IKE daemon initiated the phase 2 IKEv2 negotiation, the address is from the TSr payload.
- Otherwise, this field displays the value `n/a`.

IKED network security information (-w) primary option

The **-w** primary option is used to display network security configuration information for each active stack on the system.

See "The z/OS UNIX ipsec command IKED network security (-w) option" on page 723 for parameter descriptions.

IKED network security information (-w) primary option syntax:
For **-w** primary option syntax see "The z/OS UNIX ipsec command syntax" on page 700.

IKED network security information (-w) primary option command syntax examples:

ipsec -w display
Displays network security information of each active stack on the system.

IKED network security information (-w) primary option report examples:

`ipsec -w display`

```
CS V2R1 ipsec   Stack Name: n/a  Fri Nov 25 01:27:33 2011
Primary:  Stack NS        Function: display          Format:     detail
Source:   IKED            Scope: n/a                 TotAvail:   n/a
SystemName:  zsystem4

StackName:                    tcpcs1
ClientName:                   nsclient1
ClientAPIVersion:             2
ServerAPIVersion:             2
NSServicesSupported:          Yes
RemoteManagementSelected:     Yes
RemoteManagementEnabled:      Yes
CertificateServicesSelected:  Yes
CertificateServicesEnabled:   Yes
NSClientIPAddress:            9.42.105.88
NSClientPort:                 8801
NSServerIPAddress:            9.42.105.234
NSServerPort:                 4159
NSServerSystemName:           zsystem3
UserID:                       userxyz
ConnectionState:              connected
TimeConnectedToNSServer       Thu Sep 16 15:08:14 2004
TimeOfLastMessageToNSServer:  Mon Sep 23 06:25:50 2004
*******************************************************
StackName:                    tcpcs2
ClientName:                   n/a
ClientAPIVersion:             2
ServerAPIVersion:             2
NSServicesSupported:          No
RemoteManagementSelected:     No
RemoteManagementEnabled:      n/a
CertificateServicesSelected:  No
CertificateServicesEnabled:   n/a
NSClientIPAddress:            n/a
NSClientPort:                 n/a
NSServerIPAddress:            n/a
NSServerPort:                 n/a
NSServerSystemName:           n/a
UserID:                       n/a
ConnectionState:              n/a
TimeConnectedToNSServer       n/a
TimeOfLastMessageToNSServer:  n/a
*******************************************************

2 entries selected
```

IKED network security information (-w) primary option report field descriptions:

For the header information, see “The ipsec command report heading” on page 735.

SystemName
The name of the system on which the report is requested.

StackName
The name of the stack as defined on the host system

ClientName
The name by which the NSS server identifies the stack when it is using IPSec management services. For more information see the ClientName parameter on the NSSStackConfig statement for the IkeConfig file in the z/OS Communications Server: IP Configuration Reference.

ClientAPIVersion
The version of the NSS client API that the NSS client is using.

1 The level of NSS support that is available is z/OS version V1R9 and later

2 The level of NSS support that is available is z/OS version V1R10 and later

ServerAPIVersion
The version of the NSS server API that the NSS server is using.

1 The level of NSS support that is available is z/OS version V1R9 and later.

2 The level of NSS support that is available is z/OS version V1R10 and later.

n/a
Indicates that NSS server version information is not currently available.

NSServicesSupported
Indicates whether NSS for IPSec is supported for the stack. The value `Yes` indicates that it is supported. The value `No` indicates that it is not supported.

RemoteManagementSelected
Indicates whether the stack is configured for remote management. The value `Yes` indicates that the stack is configured for the NSS remote management service. The value `No` indicates that the stack is not configured for the NSS remote management service.

RemoteManagementEnabled
Indicates whether the stack is enabled for remote management at the NSS server. The value `Yes` indicates that the stack is permitted to access the NSS remote management service. The value `No` indicates that the stack is not permitted to access the NSS remote management service.

CertificateServicesSelected
Indicates whether the stack is configured for certificate services. The value `Yes` indicates that the stack is configured for the NSS certificate service. The value `No` indicates that the stack is not configured for the NSS certificate service.

CertificateServicesEnabled
Indicates whether the stack is enabled for certificate services at the NSS server. The value `Yes` indicates that the stack is permitted to access the NSS certificate service. The value `No` indicates that the stack is not permitted to access the NSS certificate service.

NSClientIPAddress
The IP address by which the NSS server knows the NSS client.

NSClientPort
The port by which the NSS server knows the NSS client.

NSServerIPAddress
The IP address of the NSS server to which the stack is connected.

NSServerPort
The port number of the NSS server to which the stack is connected.

NSServerSystemName
The name of the system on which the NSS server is running.

UserID
The user ID that the stack used to connect to the NSS server.

ConnectionState
The state of the connection to the NSS server. The possible states are:

connected
Indicates that the stack can use enabled network security services.

connect pending
Indicates that the stack has requested a connection to the NSS server but it is not yet connected.

update pending
Indicates that the client has dynamically reconfigured its authentication information or its requested network security services. The client has requested a connection update but has not received a successful response from the NSS server.

disconnect pending
Indicates that the stack has requested that the connection be disconnected but it is not yet disconnected.

disconnected
Indicates that the stack is not connected to the NSS server.

TimeConnectedToNSServer
The time at which the stack connected to the NSS server.

TimeOfLastMessageToNSServer
The time that the stack last received a message from the NSS server.

Network security server (-x) primary option

Use the **-x** primary option to display information about NSS IPSec clients that are currently connected to the NSS server.

See "The z/OS UNIX ipsec command network security server (-x) option" on page 723 for parameter descriptions.

Network security server (-x) primary option syntax:
For **-x** primary option syntax see "The z/OS UNIX ipsec command syntax" on page 700.

Network security server (-x) primary option command syntax examples:

ipsec -x display
Display the status of all IPSec clients that are currently connected to the NSS server.

ipsec -x display -z nsclient1
Display the status of client nsclient1 that are currently connected to the NSS server.

Network security server (-x) primary option report examples:

```
ipsec -x display

CS V2R1 ipsec   NS Client Name: n/a  Fri Nov 25 01:27:33 2011
Primary:  NS Server       Function: display         Format:     detail
Source:   Server          Scope: n/a                TotAvail:   n/a
System Name: zsystem7

ClientName:                   client1
```

```
ClientAPIVersion:            2
StackName:                   tcpcs1
SystemName:                  zsystem2
ClientIPAddress:             9.42.105.88
ClientPort:                  8801
ServerIPAddress:             9.42.105.234
ServerPort:                  4159
UserID:                      userxyz
RemoteManagementSelected:    Yes
RemoteManagementEnabled:     Yes
CertificateServicesSelected: Yes
CertificateServicesEnabled:  Yes
ConnectionState:             connected
TimeConnected:               Thu Sep 16 15:08:14 2004
TimeOfLastMessageFromClient: Mon Sep 23 06:25:50 2004
*******************************************************
ClientName:                  client2
ClientAPIVersion:            2
StackName                    tcpcs2
SystemName:                  zsystem3
ClientIPAddress:             9.42.105.88
ClientPort:                  8802
ServerIPAddress:             9.42.105.234
ServerPort:                  4159
UserID:                      userabc
RemoteManagementSelected:    No
RemoteManagementEnabled:     No
CertificateServicesSelected: Yes
CertificateServicesEnabled:  Yes
ConnectionState:             connected
TimeConnected:               Fri Sep 17 05:03:11 2004
TimeOfLastMessageFromClient: Wed Sep 21 11:25:50 2004
*******************************************************
ClientName:                  client3
ClientAPIVersion:            2
StackName                    tcpcs3
SystemName:                  zsystem4
ClientIPAddress:             9.42.105.88
ClientPort:                  8803
ServerIPAddress:             9.42.105.234
ServerPort:                  4159
UserID:                      userklm
RemoteManagementSelected:    Yes
RemoteManagementEnabled:     Yes
CertificateServicesSelected: No
CertificateServicesEnabled:  No
ConnectionState:             connected
TimeConnected:               Wed Sep 15 22:25:50 2004
TimeOfLastMessageFromClient: Mon Sep 16 12:05:33 2004
*******************************************************

3 entries selected
```

Network security server (-x) primary option report field descriptions:
For the header information, see "The ipsec command report heading" on page 735.

ClientName

The name of the NSS client.

ClientAPIVersion

The version of the NSS client API that the NSS client is using.

1 The level of NSS support that is available is z/OS version V1R9 and later.

2 The level of NSS support that is available is z/OS version V1R10 and later.

StackName
The name of the stack as defined on the client system.

SystemName
The name of the system on which the client is running.

ClientIPAddress
The IP address by which the NSS server knows the NSS client.

ClientPort
The port by which the NSS server knows the NSS client.

ServerIPAddress
The NSS server's IP address.

ServerPort
The NSS server's port number.

UserID
The user ID of the NSS client that is used to connect to the NSS server.

RemoteManagementSelected
Indicates whether the client is configured for remote management. The value `Yes` indicates that the client is configured for the NSS remote management service. The value `No` indicates that the client is not configured for the NSS remote management service.

RemoteManagementEnabled
Indicates whether the client is enabled for remote management at the NSS server. The value `Yes` indicates that the client is permitted to access the NSS remote management service. The value `No` indicates that the client is not permitted to access the NSS remote management service.

CertificateServicesSelected
Indicates whether the client is configured for certificate services. The value `Yes` indicates that the client is configured for the NSS certificate service. The value `No` indicates that the client is not configured for the NSS certificate service.

CertificateServicesEnabled
Indicates whether the client is enabled for certificate services at the NSS server. The value `Yes` indicates that the client is permitted to access the NSS certificate service. The value `No` indicates that the client is not permitted to access the NSS certificate service.

ConnectionState
The state of the connection from the NSS client. The possible states are:

connected
Indicates that the client can use enabled NSS services

connect pending
Indicates that the client has requested a connection to the NSS server but it is not yet connected.

update pending
Indicates that the client has dynamically reconfigured its authentication information or its requested NSS services. The client has requested a connection update but has not received a successful response from the NSS server.

disconnect pending
Indicates that the client has requested that the connection be disconnected but it is not yet disconnected.

TimeConnected
The time at which the NSS client connected to the NSS server.

TimeOfLastMessageFromClient
The time that the NSS server last received a message from the client.

nssctl command

Use the z/OS UNIX **nssctl** command to display information for NSS clients that are currently connected to the local NSS server. For a description of the terms and concepts that are used in the **nssctl** command information, see IP security and network security services information in z/OS Communications Server: IP Configuration Guide.

You can use the **nssctl** command to display information that is maintained by the NSS server for all NSS clients that are currently connected to the NSS server. You can also display information for a specific NSS client or for clients that use a specific NSS discipline.

nssctl command security

The **nssctl** command is an APF-authorized application. Users of the **nssctl** command must be authorized through the security access facility (SAF). This information assumes that the SAF that is being used is RACF. User authorization is managed with the SERVAUTH profile and is described in "nssctl command SERVAUTH profile."

nssctl command SERVAUTH profile

Security product authorization (for example, RACF) is required to use the **nssctl** command. You must define a profile in the SERVAUTH class to enable control over the **nssctl** command function. The format of the profile is as follows:

```
EZB.NETMGMT.sysname.sysname.NSS.DISPLAY
```

Where:

sysname
The system name of the system on which the **nssctl** command is allowed to run.

Requirement: This profile must be defined on the system where the NSS server is running and where the **nssctl** command is issued.

Tips:
- If the security product is RACF, you can use the control statements in the sample JCL job that is provided in SEZAINST(EZARACF) to define this authorization. If the SERVAUTH class is not active or if a matching SERVAUTH policy is not found, the **nssctl** command request is rejected.
- Authorization is not required for the help option (**nssctl -?**).

The z/OS UNIX nssctl command syntax

The z/OS UNIX **nssctl** command displays information for NSS clients that are currently connected to the local NSS server.

Restriction: The **nssctl** command must be issued on the same host z/OS system as the NSS server.

Format

```
►►──nssctl──┤ Primary Option ├──┤ Debug Option ├────────────────────────────►◄
```

Primary Option:

```
├──┬─ -d─┤ Filter Option ├─┬────────────────────────────────────────────────┤
   └─ -?───────────────────┘
```

Filter Option:

```
├──┬─ -c── nsclientname────┬────────────────────────────────────────────────┤
   └─ -D──┬─ipsec────────┬─┘
          └─xmlappliance─┘
```

Debug Option:

```
          ┌─3──────────────┐
├── -Z────┴───debuglevel───┴──────────────────────────────────────────────────┤
```

The z/OS UNIX nssctl command parameter descriptions

The following information describes the individual parameter items that are identified in the syntax diagram. All options are case sensitive. Option values that are keywords and the *nsclientname* value for **-c** filter option are not case sensitive.

The z/OS UNIX nssctl command primary options

-d Display NSS server information for each NSS client.

-? Display command help.

The z/OS UNIX nssctl command filter options

The following parameters can be used to filter the output of the specified report.

-c *nsclientname*
Filter the output of the display report using the specified NSS client name *nsclientname*.

-D Filter the report output using the specified discipline type.

ipsec
Filter the output of the display report to display NSS server information only for NSS IPSec clients.

xmlappliance
Filter the output of the display report to display NSS server information only for NSS XMLAppliance clients. The xmlappliance keyword can be shortened to the first four characters, for example, you can specify **-D xmlappliance** as **-D xmla**.

The z/OS UNIX nssctl command debug options

-Z Generates debug information when the command runs. An optional *debuglevel* value can be specified with the **-Z** option. Debug output is sent to stderr or to stdout, as determined by the debug level. Debug information for a particular debug level also includes the information from lower levels (for example, debug level 3 also includes the information from level 1 and level 2).

debuglevel
The following debug levels are available:

1 Generate functional level debug information in stdout in a formatted form.

2 Generate general debug information in stderr in an unformatted form for criteria that was specified when the command was issued.

3 Generate operational level debug information in stderr in an unformatted form. This is the default debug level.

The nssctl command report details and examples

This information contains descriptions about the formatting and contents of **nssctl** reports, including examples.

The nssctl command general report concepts

To fully understand the following concepts and fields, you need some general knowledge of NSS. See information about preparing to provide network security services in z/OS Communications Server: IP Configuration Guide for more details.

The nssctl command report heading:

The display report from the **nssctl** command begins with several heading lines that provide general information related to the request.

The nssctl command heading example:

```
CS V2R1 nssctl   System Name: zsystem7  Tue Feb 14 14:48:50 2012
Function: display       NS Client Name: n/a
```

The first heading line displays the following information:

System Name
The system name of the system on which the server is running.

timestamp
The date and time of the report.

The second heading line displays the following information:

Function
The function option for any report is display.

NS Client Name
The name of the NSS client when the **-c** option is used. Otherwise, the value is n/a.

The final line of the display report shows how many entries were actually listed in the report. Depending on the selection criteria that was specified on the request, the count of entries in the report might be less than the entire set that is being queried.

The nssctl command report data:

All data fields are shown, even if some of the fields are not applicable to the type of entry that is being displayed or if some of the fields are not applicable to the context of the data. Data fields in each entry display the common information first, followed by the discipline-specific information.

nssctl -d report

Display information about NSS clients that are managed by the NSS server.

nssctl -d report command syntax:

For -d primary option syntax see "The z/OS UNIX nssctl command syntax" on page 808.

nssctl -d report command syntax examples:

nssctl -d
Display the status of all clients that are connected to the local NSS server.

nssctl -d -c nsclient1
Display the status of the nsclient1 client that is connected to the local NSS server.

nssctl -d report example:

nssctl -d:

```
CS V2R1 nssctl  SystemName: MVS093    Sat Dec 10 08:47:39 2011
Function: Display            NSSClientName: n/a

ClientName:                     clientIB1
ClientAPIVersion:               2
StackName:                      tcpcs1
SystemName:                     zsystem2
ClientIPAddress:                9.42.105.88
ClientPort:                     8801
ServerIPAddress:                9.42.105.234
ServerPort:                     4159
UserID:                         userxyz
ConnectState:                   connected
TimeConnected:                  2011/12/10 08:47:05
TimeOfLastMessageFromClient:    2011/12/10 08:47:05
Discipline:                     IPSec
  CertificateServiceSelected:   Yes
  CertificateServiceEnabled:    Yes
  RemoteManagementSelected:     Yes
  RemoteManagementEnabled:      Yes
***************************************************************
ClientName:                     ClientXB1
ClientAPIVersion:               3
StackName:                      Any
SystemName:                     dpsys01
ClientIPAddress:                ::ffff:10.11.1.5
ClientPort:                     1024
ServerIPAddress:                ::ffff:10.81.1.1
ServerPort:                     4159
UserID:                         USER5
ConnectState:                   connected
TimeConnected:                  2011/12/10 08:47:13
TimeOfLastMessageFromClient:    2011/12/10 08:47:13
Discipline:                     XMLAppliance
  CertificateServiceSelected:   Yes
  CertificateServiceEnabled:    Yes
  PrivateKeyServiceSelected:     Yes
  PrivateKeyServiceEnabled:      Yes
  SAFAccessServiceSelected:      Yes
  SAFAccessServiceEnabled:       Yes
****************************************************************

2 entries selected
```

nssctl -d report field descriptions:

For the header information, see "The nssctl command report heading" on page 810.

`ClientName`
The name of the NSS client. If the client is not authenticated to the server, a temporary client name is displayed.

`ClientAPIVersion`
The version of the NSS client API that the NSS client is using. Possible values are.

1 The level of NSS support that is available in z/OS V1R9 and later.

2 The level of NSS support that is available in z/OS V1R10 and later.

3 The level of NSS support that is available in z/OS V1R11 and later.

`StackName`
The name of the stack as defined on the client system. If the NSS client is not running on a z/OS system, then this value is not applicable.

`SystemName`
The name of the system on which the client is running.

`ClientIPAddress`
The IP address by which the NSS server knows the NSS client.

`ClientPort`
The port by which the NSS server knows the NSS client.

`ServerIPAddress`
The IP address of the NSS server.

`ServerPort`
The port number of the NSS server.

`UserID`
The user ID of the NSS client that is used to connect to the NSS server.

`ConnectionState`
The state of the connection from the NSS client. Possible state values are:

`connected`
Indicates that the client can use enabled NSS services.

`connect pending`
Indicates that the client has requested a connection to the NSS server but it is not yet connected.

`update pending`
Indicates that the client has dynamically reconfigured its authentication information or its requested NSS services. The client has requested a connection update but has not received a successful response from the NSS server.

disconnect pending
Indicates that the client has requested to disconnect the connection, but the connection is not yet disconnected.

TimeConnected
The time at which the NSS client connected to the NSS server.

TimeOfLastMessageFromClient
The time that the NSS server last received a message from the client.

Discipline
The discipline from which the client is requesting NSS services. Possible discipline values are:

IPSec
The IPSec client is requesting NSS services. The following IPSec discipline-specific information is displayed:

CertificateServiceSelected
Indicates whether the IPSec client is configured for the IPSec certificate service. The value Yes indicates that the IPSec client is configured for the NSS IPSec certificate service. The value No indicates that the IPSec client is not configured for the NSS IPSec certificate service.

CertificateServiceEnabled
Indicates whether the IPSec client is enabled for the IPSec certificate service at the NSS server. The value Yes indicates that the IPSec client is requested and permitted to access the NSS IPSec certificate service. The value No indicates that the IPSec client is not permitted to access the NSS IPSec certificate service.

RemoteManagementSelected
Indicates whether the IPSec client is configured for remote management. The value Yes indicates that the IPSec client is configured for the NSS remote management service. The value No indicates that the IPSec client is not configured for the NSS remote management service.

RemoteManagementEnabled
Indicates whether the IPSec client is enabled for remote management at the NSS server. The value Yes indicates that the IPSec client is permitted to access the NSS remote management service. The value No indicates that the IPSec client is not permitted to access the NSS remote management service.

XMLAppliance
The XMLAppliance client is requesting NSS services. The following XMLAppliance discipline-specific information is displayed:

CertificateServiceSelected
Indicates whether the XMLAppliance client requested the XMLAppliance certificate service. The value Yes indicates that the XMLAppliance client requested the NSS XMLAppliance certificate service. The value No indicates that the XMLAppliance client did not request the NSS XMLAppliance certificate service.

CertificateServiceEnabled
Indicates whether the certificate service is enabled for the XMLAppliance client. The value Yes indicates that the certificate

service is requested and permitted for the XMLAppliance client. The value `No` indicates that the certificate service is not permitted for the XMLAppliance client.

`PrivateKeyServiceSelected`
Indicates whether the XMLAppliance client requested the private key service. The value `Yes` indicates that the XMLAppliance client requested the NSS private key service. The value `No` indicates that the XMLAppliance client did not request the NSS private key service.

`PrivateKeyServiceEnabled`
Indicates whether the private key service is enabled for the XMLAppliance client. The value `Yes` indicates that the private key service is requested and permitted for the XMLAppliance client. The value `No` indicates that the private key service is not permitted for the XMLAppliance client.

`SAFAccessServiceSelected`
Indicates whether the XMLAppliance client requested the SAF access service. The value `Yes` indicates that the XMLAppliance client requested the NSS SAF access service. The value `No` indicates that the XMLAppliance client did not request the NSS SAF access service.

`SAFAccessServiceEnabled`
Indicates whether the SAF access service is enabled for the XMLAppliance client. The value `Yes` indicates that the SAF access service is requested and permitted for the XMLAppliance client. The value `No` indicates that the SAF access service is not permitted for the XMLAppliance client.

certbundle command

Use the z/OS UNIX **`certbundle`** command to create a certificate bundle file. A certificate bundle file can contain certificate and certificate revocation list (CRL) information. You can place certificate bundle files on an HTTP server. You can configure the CertificateBundleURL parameter on the IPSecDisciplineConfig statement in the configuration file of the network security services server to associate a label on the key ring of the NSSD with the URL for the certificate bundle. See the information about storing certificate-related information on an HTTP server in z/OS Communications Server: IP Configuration Guide for additional details.

The z/OS UNIX certbundle command syntax

Format

```
►►──certbundle──┤ Primary Option ├──┤ Debug Option ├──────────────────────────►◄
```

Primary Option:

```
├──┬─ -c─┤ Create Option ├─┬─────────────────────────────────────────────────┤
   └─ -?───────────────────┘
```

Create Option:

```
├── -i cert_bundle_options_file─────────────────────────────────────────────┤
```

Debug Option:

```
            ┌ 3──────────┐
├── -d──────┤            ├──────────────────────────────────────────────────┤
            └ debuglevel─┘
```

The z/OS UNIX certbundle command parameter descriptions

The following topics describe the individual parameter items that appear in the syntax diagram. All options are case sensitive.

The z/OS UNIX certbundle command primary options

-c Create certificate bundle files.

-? Display command help.

The z/OS UNIX certbundle command create options

-i *cert_bundle_options_file*
Name of the input certificate bundle options file. See "The z/OS UNIX certbundle command options file" for more information.

The z/OS UNIX certbundle command debug options

-d

Generates debug information during command processing. You can specify an optional *debuglevel* value when you use the **-d** option. The debug level determines the level of debug information output to stdout. The debug information accumulates as the level number increases; for example, debug level 3 also includes the information from level 1 and level 2.

debuglevel
The following debug levels are available:

1 Generate functional-level debug information in stdout in a formatted form.

2 Generate additional information in stdout in a formatted form. Example additional information includes key ring and certificate operations.

3 Generate verbose information in stdout in a formatted form. Example verbose information includes file parsing and memory management. This value is the default debug level.

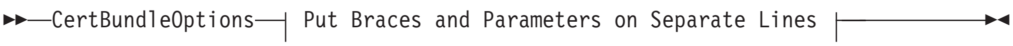
The z/OS UNIX certbundle command options file

The certificate bundle options file identifies the location of certificates and certificate revocation that are to be included in a certificate bundle.

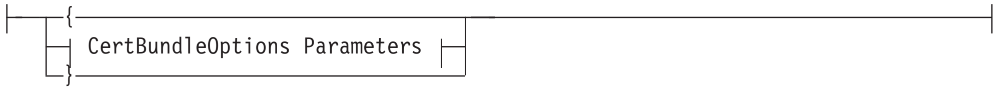
CertBundleOptions statement

You can code more than one CertBundleOptions statement. Each CertBundleOptions statement is processed separately.

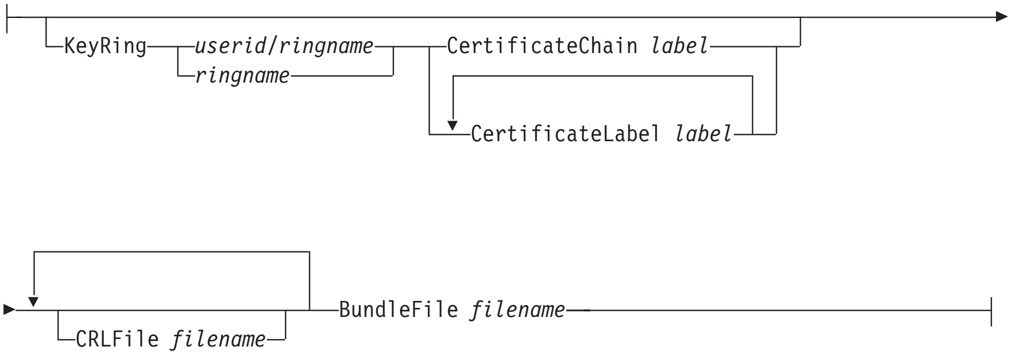
Format



Put Braces and Parameters on Separate Lines:



CertBundleOptions Parameters:



Parameters

KeyRing *ringname* **|** *userid/ringname*
The owning user ID and ring name to be used. When you specify a key ring that is owned by the issuer of the command, specify the ring name as the *ringname* value. When you specify a key ring owned by another user, specify the ring name as a *userid/ringname* value. Specification of the KeyRing parameter requires specification of either the CertificateChain or CertificateLabel parameter. There is no default value.

CertificateChain *label*
The label of a certificate on the SAF key ring. This certificate and all certificates that are in the trust chain of this certificate that are in the key ring (excluding the certificate of the root certificate authority) are included in the bundle. The KeyRing parameter must also be specified to locate the specified certificate label.

CertificateLabel *label*
The label of a certificate on the SAF key ring. This certificate is included in the bundle. The KeyRing parameter must also be specified to locate the specified certificate label.

CRLFile *filename*
The filename of the certificate revocation list file.

Restriction: When you generate a certificate bundle with CRLFile only, the KeyRing parameter must not be specified.

BundleFile *filename*
The name of the output certificate bundle file to be created. If this file already exists, it is replaced by the new file.

Examples

The following certificate bundle options file creates three separate certificate bundle files:

- The first bundle file (/etc/security/BundleFile1) contains the three specified certificates from mykeyring and the CRL from the file /etc/security/CRL1.
- The second bundle file (/etc/security/BundleFile2) contains only the CRL from the file /etc/security/CRL2.
- The third bundle file (/etc/security/BundleFile3) contains only the three specified certificates from mykeyring.

```
CertBundleOptions
{
  Keyring   mykeyring
   CertificateLabel  label1
   CertificateLabel  label2
   CertificateLabel  label3
  CRLFile   /etc/security/CRL1
  BundleFile        /etc/security/BundleFile1
}
CertBundleOptions
{
  CRLFile   /etc/security/CRL2
  BundleFile        /etc/security/BundleFile2
}
CertBundleOptions
{
  Keyring   NSSD/mykeyring
  CertificateLabel  labelA
   CertificateLabel  labelB
   CertificateLabel  labelC
  BundleFile        /etc/security/BundleFile3
}
```

Chapter 5. Displaying policy-based networking information

You can use the following TCP/IP commands to display policy-based networking information from the network.

- The z/OS UNIX **pasearch** command queries information from the z/OS UNIX Policy Agent.
- The z/OS UNIX **trmdstat** command displays the Traffic Regulation Management Daemon (TRMD) log.

See Chapter 3, “Monitoring the TCP/IP network,” on page 303 for Netstat commands, such as Netstat SLAP/-j or Netstat IDS/-k, for information that might be relevant to retrieving information from the network, and “ipsec command security” on page 696 for information about the **ipsec** command.

Additionally, you can monitor policy implementation using the Network SLAPM2 subagent. Using SNMP, you can display policy configuration and performance data and generate notifications when monitored traffic performance crosses thresholds defined in the Network SLAPM2 MIB tables. See information about the Network SLAPM2 subagent in z/OS Communications Server: IP Configuration Guide for more details about using SNMP to monitor policy performance.

The z/OS UNIX pasearch command: Display policies

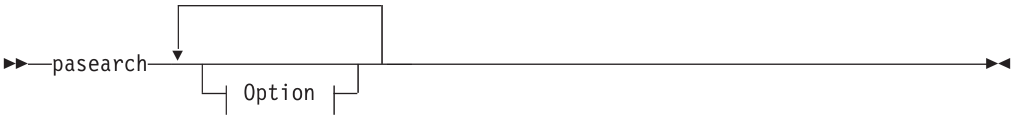
Use the z/OS UNIX **pasearch** command to query information from the z/OS UNIX Policy Agent. The command is issued from the UNIX System Services shell.

Restriction: The **pasearch** command requires access to the PAPI DLL at run time. Ensure that the LIBPATH environment variable is specified and points to the /usr/lib directory. For example, specify: export LIBPATH=/usr/lib

Note: If the user is *not* a superuser, see z/OS Communications Server: IP Configuration Guide for information about configuring the Policy Agent and setting up authorization for the client to retrieve policies.

Result: If any of the information that is requested by the **pasearch** command is not currently available, the **pasearch** command displays `<not available>`. For example, when the **pasearch** command is issued on a policy client, some information might need to be obtained from the policy server. Reissue the **pasearch** command later to see the complete information.

Format



Option:

```
    ┌─ -A -e──────────────────┐
├───┼─────────────────────────┼──────────────────────────────────────────────┤
    ├─ -A─────────────────────┤
    ├─ -a─────────────────────┤
    ├─ -C─────────────────────┤
    ├─ -c─────────────────────┤
    ├─ -d─────────────────────┤
    ├─ -e─────────────────────┤
    ├─ -f ──PolicyFilterName──┤
    ├─ -g─────────────────────┤
    ├─ -I─────────────────────┤
    ├─ -i─────────────────────┤
    ├─ -n─────────────────────┤
    ├─ -o─────────────────────┤
    ├─ -p ──image─────────────┤
    ├─ -q─────────────────────┤
    ├─ -R─────────────────────┤
    ├─ -r─────────────────────┤
    ├─ -s ──PolicyScopeName───┤
    ├─ -T─────────────────────┤
    ├─ -t─────────────────────┤
    ├─ -v───┬─a─┬─────────────┤
    │       ├─f─┤             │
    │       ├─k─┤             │
    │       └─l─┘             │
    ├─ -w─────────────────────┤
    └─ -?─────────────────────┘
```

Parameters

-A Display active policy entries that match input options for **pasearch**. This is the default. If all policy entries are requested (**pasearch -e**, **pasearch**, or **pasearch -a -r**) and the policy rule is active, then active policy actions are returned. Policies on the policy server that are loaded on behalf of policy clients always display as active policies.

-a Display all policy actions that match the input options for the **pasearch** command. Because the default action is to return all types of policy actions, use the **-i**, **-q**, **-R**, **-t**, or **-v** option to limit the type of policy actions that are returned.

-C Display all image names with policies that are configured in Policy Agent. This includes locally defined images (those defined on a TcpImage statement) and connected policy clients (where the image name is defined by each client on the *ClientName* parameter on the PolicyServer statement).

-c Display policy object information (for example, FLUSH or NOFLUSH, PURGE or NOPURGE). This option can be used with the image option (**-p**), or the policy type options (**-i**, **-q**, **-R**, **-t**, or **-v**). All other options are either ignored or are not valid.

See the following descriptions of policy object fields:

ConfigLocation

Indicates the source from which the policies were loaded. The following might be displayed on the policy server:

Local Indicates that the policies were loaded from local configuration files, an LDAP server, or both.

Client Indicates that the policies were loaded for a connected policy client.

The following might be displayed on the policy client:

Local Indicates that the policies were loaded from local configuration files, an LDAP server, or both.

Remote
Indicates that the policies were loaded from the policy server.

LDAPServer
Indicates whether or not an LDAP server is used for local policies.

CommonFileName
Indicates the name of the common configuration file, if one exists.

ImageFileName
Indicates the name of the stack-specific configuration file.

ClientName
Indicates the policy client name.

ClientUserid
Indicates the user ID being used for a policy client.

PolicyServerAddr
Indicates the IP address of the policy server being used for remote policies.

PolicyServerPort
Indicates the port of the policy server being used for remote policies.

PolicyServSysname
Indicates the system name of the policy server being used for remote policies.

PolicyClientAddr
Indicates the IP address of a connected policy client.

PolicyClientPort
Indicates the port of a connected policy client.

ConnectTime
Indicates the time when a policy client connected to the policy server.

ApplyFlush
Indicates whether the policy type uses the PolicyFlush flag for FLUSH or NOFLUSH processing.

DeleteOnNoflush
Indicates whether or not NOFLUSH processing is honored.

ApplyPurge
Indicates whether the policy type uses the PurgePolicies flag for PURGE or NOPURGE processing.

AtomicParse
Indicates whether or not parsing of the policy type is atomic. With atomic parsing, any errors result in the entire set of policy changes for that policy type being discarded. Without atomic parsing, only objects found to be in error are discarded.

DummyOnEmptyPolicy
Indicates whether the TCP/IP stack is informed if no policies are configured for this type of policy.

ModifyOnIDChange
Indicates whether or not a rule or action object is considered changed if only the rule or action ID changes due to the order of policies.

PolicyFlush
For policy types that honor FLUSH, indicates whether FLUSH or NOFLUSH was configured on the TcpImage, PEPInstance, or specific type configuration statement (for example TTLSConfig).

PurgePolicies
For policy types that honor PURGE, indicates whether PURGE or NOPURGE was configured on the TcpImage, PEPInstance, or specific type configuration statement (for example TTLSConfig).

Configured
Indicates whether any policies were configured for this policy type.

UpdateInterval
Indicates the time interval (in seconds) for checking the creation or modification time of the configuration file or files, and for refreshing policies from the LDAP server.

PerfColEnabled
Indicates whether the PolicyPerformanceCollection statement was enabled.

InstanceId
An identification associated with the last update for this policy type.

LastPolicyChanged
The time stamp value that indicates when any policy rule, policy action, or table for this policy type was last updated.

Policy updated
The time stamp value that indicates when the IPSec policy object was last updated.

`-d` Display debug information to stdout.

`-e` Display all policy entries (policy rules and policy actions) that match the input options for the **pasearch** command. If policy action matches, then the associated policy rule is returned. This is the default.

`-f` *`PolicyFilterName`*
Display policy entries that match the policy name based on input options for the **pasearch** command. For a policy rule or policy action the name is either the policy name specified on the configuration file statement that defines the policy entry (policy rule or policy action) or the name specified using the *ServiceName*, *policyActionName*, *PolicyRulesName*, or *policyRuleName* attribute for policy entries defined on an LDAP server. For the route table the name is the name configured on the RouteTable statement.

Rules:

- The name is case sensitive.
- To match the *PolicyFilterName* attribute with multiple policy entries, use the **-w** option with the **-f** option. The *PolicyFilterName* attribute is treated as a wildcard name; the default action is to find an exact match.

- To match the *PolicyFilterName* attribute with the policy rule name, do not use the **-g** option with the **-f** option. This is the default.
- To match the *PolicyFilterName* attribute with the policy action name, use the **-g** option with the **-f** option.
- To match the *PolicyFilterName* attribute with the route table name, use the **-T** option with the **-f** option.

-g Matches the *PolicyFilterName* attribute to policy actions. If retrieving both policy rules and policy actions, then this request returns a policy rule when there is a matching policy action. If no *PolicyFilterName* attribute is passed, then no action name filtering is performed.

-I Display inactive policy entries that match input options for the **pasearch** command. If all policy entries are requested (**pasearch -e -I**, **pasearch -I**, or **pasearch -I -a -r**) and the policy rule is inactive, then inactive policy actions are returned. Policies on the policy server that are loaded on behalf of policy clients always display as active policies.

-i Display all IDS policy entries that match the input options for the **pasearch** command.

-n Display only policy rule, policy action, or route table names (policy details are not displayed).

-o Display the policy rule condition original level and condition original arrays. This option applies only to complex rules (those that use CNF or DNF conditions). For such rules, there are two sets of condition arrays maintained: the original set of specified conditions, and a working set that has been collapsed or summarized for performance reasons. By default, only the working set is displayed. Use this option to display the original set.

-p *image*

Display all policy entries that belong to the specified *image* name that match input options for the **pasearch** command. The default action is to return all policy entries for all TCP/IP stacks. The value used for the *image* name must match one of the values that is specified on the TcpImage or PEPInstance statement in the Policy Agent configuration file, or match a connected policy client name.

Result: If the **-p** option is not used, then only the policies that are configured with the TcpImage or PEPInstance statement are returned.

-q Display all QoS policy entries that match the input options for the **pasearch** command.

-R Display all Routing policy entries that match the input options for the **pasearch** command.

- With the **-e** option, this displays Routing policy rules and policy actions. This is the default.
- With the **-r** option or the **-a** option, this displays Routing policy rules or policy actions.
- With the **-T** option, this displays route tables.

-r Display all policy rules that match the input options for the **pasearch** command.

-s *PolicyScopeName*

Display all policy actions that match the *PolicyScopeName* value. The *PolicyScopeName* attribute is not case sensitive.

- Display all QoS, IpFilter, or AT-TLS policy actions that match the *PolicyScopeName* value.
  - Valid QoS *PolicyScopeName* values are DataTraffic, RSVP, or both.
  - Valid IpFilter *PolicyScopeName* values are DynamicVpn, ManualVpn, GenericFilter, or LocalStart.
  - Valid AT-TLS *PolicyScopeName* values are Group, Environment, or Connection.
- If both policy rules and policy actions are requested (**pasearch -e -s** *PolicyScopeName* or **pasearch -a - r -s** *PolicyScopeName*), then the policy rule is returned with all its policy actions when there is a matching policy action with the requested *PolicyScopeName* value.

-T Display all tables that match the input options for the **pasearch** command. The only supported table is routing policy type (**-R**). The **-R** policy type is the default.

- With the **-A** option, the **-T** option displays active routing tables. These are routing tables that are configured and referenced by an active Routing policy rule and its associated Routing policy action. This is the default.
- With the **-I** option, the **-T** option displays inactive routing tables. These are routing tables that are configured but not referenced by an active Routing policy rule and its associated Routing policy action.

-t Display all Application Transparent Transport Layer Security (AT-TLS) policy entries that match the input options for **pasearch**.

Results:

- Pasearch does not display optional parameters that do not have a default value.
- Pasearch does not display the value of a password parameter and indicates only whether it is configured with a value of Yes or No.

-v

Displays IPSec IpFilter, KeyExchange, and LocalDynVpn policies that match the input options for the **pasearch** command.

a Display all IPSec policy entries.

f Display only IpFilter policy entries.

k Display only KeyExchange policy entries.

l Display only LocalDynVpn policy entries.

-w The *PolicyFilterName* is a wildcard to be matched to the name. For example, if *PolicyFilterName* = Web, then all policy rules, policy actions, or route tables with the first 3 characters of their names equal to Web are returned. If no *PolicyFilterName* is passed, then no name filtering is done.

-? Display **pasearch** options help information.

Examples

The following example shows policy object information for all types of policies:

```
========================================================================
================== pasearch -c =========================================
========================================================================

TCP/IP pasearch CS V2R1                 Image Name: TCPCS1
  Date:                09/18/2011        Time:  13:30:32
  PAPI Version:        9                 DLL Version:  9
```

```
Qos Policy Object:
  ConfigLocation:       Local             LDAPServer:        True
  ImageFileName:        /u/user10/pagallcimagea.conf
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     True
  AtomicParse:          False             DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   False             ModifyOnIDChange:  True
  Configured:           True              UpdateInterval:    120
  PerfColEnabled:       False
  InstanceId:           1253294875
  LastPolicyChanged:    Fri Sep 18 13:27:55 2011

Ids Policy Object:
  ConfigLocation:       Local             LDAPServer:        True
  CommonFileName:
  ImageFileName:        /usr/lpp/tcpip/samples/pagent_IDS.conf
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     True
  AtomicParse:          False             DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   False             ModifyOnIDChange:  False
  Configured:           True              UpdateInterval:    120
  InstanceId:           1253294875
  LastPolicyChanged:    Fri Sep 18 13:27:55 2011

IPSec Policy Object:
  ConfigLocation:       Remote            LDAPServer:        False
  ClientName:           VIC136_TCPCS1
  ClientUserid:         USER1
  PolicyServerAddr      9.42.104.23
  PolicyServerPort:     8211              PolicyServSysname: VIC137
  ClientSSLActive:      True
  ConnectTime:          Fri Sep 18 13:29:51 2011
  ApplyFlush:           False
  ApplyPurge:           False
  AtomicParse:          True              DeleteOnNoflush:   True
  DummyOnEmptyPolicy:   True              ModifyOnIDChange:  False
  IpSecEnabled IPv4:    True              IpSecEnabled IPv6: False
  IpSec3DESEnabled:     True              IpSecAESEnabled:   True
  IpSecAESGCM16Enabled: True
  UpdateInterval:       300
  InstanceId:           1253294993
  LastPolicyChanged:    Fri Sep 18 13:29:53 2011
  IpFilter Policy Object:
   Configured:          True              PreDecapOn:        Off
   FilterLogging:       On                FilterLogImplicit: No
   AllowOnDemand:       No                ImplDiscardAction: Silent
   FIPS140:             No
  KeyExchange Policy Object:
   Configured:          True
   AllowNat:            No                NatKeepAliveIntvl: 20
   HowToInitiate:       Main              LivenessInterval:  30
   BypassIpValidation:  No                CertURLLookupPref: Tolerate
   RevocationChecking:  Loose
  LocalDynVpn Policy Object:
   Configured:          True
  Policy updated:       Fri Sep 18 13:29:53 2011

Routing Policy Object:
  ConfigLocation:       Local             LDAPServer:        False
  CommonFileName:
  ImageFileName:        /usr/lpp/tcpip/samples/pagent_Routing.conf
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     False
  AtomicParse:          True              DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   True              ModifyOnIDChange:  False
  Configured:           True              UpdateInterval:    120
  InstanceId:           1253294871
  LastPolicyChanged:    Fri Sep 18 13:27:51 2011

TTLS Policy Object:
  ConfigLocation:       Remote            LDAPServer:        False
  ClientName:           VIC136_TCPCS1
  ClientUserid:         USER1
  PolicyServerAddr      9.42.104.23
  PolicyServerPort:     8211              PolicyServSysname: VIC137
  ClientSSLActive:      True
  ConnectTime:          Fri Sep 18 13:29:51 2011
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     True
  AtomicParse:          True              DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   True              ModifyOnIDChange:  False
  Configured:           True              UpdateInterval:    300
  TTLS Enabled:         False
  InstanceId:           1253294993
  LastPolicyChanged:    Fri Sep 18 13:29:53 2011
```

The following example shows active QoS policies for TCP image TCPCS:

```
========================================================================
================== pasearch -q -p TCPCS1 ===============================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS1
  Date:                 09/18/2011        Time:  13:30:32
  QoS Instance Id:      1253294875

policyRule:             web-catalog-rule
  Rule Type:            QoS
  Version:              3                 Status:            Active
  Distinguish Name:     cn=web-catalog-rule,cn=QoS,cn=advanced,ou=policy,o=IBM,c=US
  Group Distinguish Nm: cn=main,cn=QoS,cn=advanced,ou=policy,o=IBM,c=US
  Weight:               110               ForLoadDist:       False
  Priority:             10                Sequence Actions:  Don't Care
  No. Policy Action:    1                 ConditionListType: DNF
  policyAction:         interactive1-action
   ActionType:          QOS
   Action Sequence:     1
  Time Periods:
   Day of Month Mask:
   First to Last:       1111111111111111111111111111111
   Last to First:       1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    1111111  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00             To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00             To TimeOfDay UTC:  04:00
   TimeZone:            Local
  Net Condition Summary:                  NegativeIndicator: Off
   RouteCondition:
    InInterface:        All
    OutInterface:       All
    IncomingTOS:        00000000          IncomingTOSMask:   0
   HostCondition:
    SourceIpFrom:       All
    SourceIpTo:         All
    DestIpFrom:         All
    DestIpTo:           All
    DestHostDomainName:
   ApplicationCondition:
    ProtocolNumFrom:    6                 ProtocolNumTo:     6
    SourcePortFrom:     80                SourcePortTo:      80
    DestPortFrom:       0                 DestPortTo:        0
    ApplicationName:                      ApplPriority:      0
    ApplicationData:    /catalog
  Policy created: Fri Sep 18 13:27:55 2011
  Policy updated: Fri Sep 18 13:27:55 2011

  Qos Action:           interactive1-action
    Version:            3                 Status:            Active
    Distinguish Name:   cn=interactive1,cn=QoSact,cn=repository,o=IBM,c=US
    Scope:              DataTraffic       OutgoingTOS:       10000000
    Permission:         Allowed
    MaxRate:            0                 MinRate:           0
    MaxConn:            0
    Routing Interfaces: 0
    RSVP Attributes:
     ServiceType:       0                 MaxRatePerFlow:    0
     MaxTokBuckPerFlw:  0                 MaxFlows:          0
     SignalClient:      True
    DiffServ Attributes:
     InProfRate:        0                 InProfPeakRate:    0
     InProfTokBuck:     0                 InProfMaxPackSz:   0
     OutProfXmtTOSByte: 00000000          ExcessTrafficTr:   BestEffort
    Policy created: Fri Sep 18 13:27:55 2011
    Policy updated: Fri Sep 18 13:27:55 2011
```

The following example shows active KeyExchange policies:

```
========================================================================
================== pasearch -v k =======================================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS1
  Date:                 09/18/2011        Time:  13:30:32
  IPSec Instance Id:    1253294993

policyRule:             Admin_KeyExRule1
  Rule Type:            KeyExchange
  Version:              3                 Status:            Active
  Weight:               105               ForLoadDist:       False
  Priority:             5                 Sequence Actions:  Don't Care
  No. Policy Action:    1
  IpSecType:            policyKeyExchange
  policyAction:         Bronze-PSK
   ActionType:          KeyExchange
   Action Sequence:     0
  Time Periods:
   Day of Month Mask:   0000000000000000000000000000000
   Month of Yr Mask:    000000000000
   Day of Week Mask:    0000000  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00             To TimeOfDay:      00:00
   Fr TimeOfDay UTC:    00:00             To TimeOfDay UTC:  00:00
   TimeZone:            Local
  IpSec Condition Summary:                NegativeIndicator: Off
   KeyExchange Condition:
    LocalSecurityEndPoint:
     Location:
      FromAddr:         All4
      ToAddr:           All4
     Identity:
      UserAtFqdn:
       admin@secureserver.raleigh.ibm.com
    RemoteSecurityEndPoint:
     Location:
      FromAddr:         9.1.1.2
      ToAddr:           9.1.1.2
     Identity:
      IpAddr:
       FromAddr:        9.1.1.2
       ToAddr:          9.1.1.2
  Policy created: Fri Sep 18 13:29:53 2011
  Policy updated: Fri Sep 18 13:29:53 2011

  KeyExchange Action:   Bronze-PSK
    Version:            3                 Status:            Active
    HowToInitiate:      Aggressive        HowToRespondIKEv1: Aggressive
    AllowNat:           No                FilterByIdentity:  No
    HowToAuthMe:        RsaSignature      ReauthInterval:    0
    BypassIpValidation: No                CertURLLookupPref: Tolerate
    KeyExchangeOffer:   0
     HowToEncrypt:      DES               KeyLength:         N/A
     HowToAuthPeers:    PresharedKey      DHGroup:           Group1
     HowToAuthMsgs:     SHA1
     HowToVerifyMsgs:   HMAC_SHA1_96      PseudoRandomFunc:  HMAC_SHA1
     RefLifeTmPropose:  480
     RefLifeTmAcptMin:  240               RefLifeTmAcptMax:  1440
     RefLifeSzPropose:  None
     RefLifeSzAccept :  None
    Policy created: Fri Sep 18 13:29:53 2011
    Policy updated: Fri Sep 18 13:29:53 2011
```

The following example shows an active LocalDynVpn policy rule:

```
========================================================================
================== pasearch -v l =======================================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS1
  Date:                 09/18/2011        Time:  13:30:32
  IPSec Instance Id:    1253294993

policyRule:             ZoneC_VPN-EE1
  Rule Type:            LocalDynVpn
  Version:              3                 Status:            Active
  GroupName:            ZoneC_BranchOfficeVPNs
  Weight:               108               ForLoadDist:       False
  Priority:             8                 Sequence Actions:  Don't Care
  No. Policy Action:    0
  IpSecType:            policyDynamicVpn
  Time Periods:
   Day of Month Mask:   0000000000000000000000000000000
   Month of Yr Mask:    000000000000
   Day of Week Mask:    0000000  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00             To TimeOfDay:      00:00
   Fr TimeOfDay UTC:    00:00             To TimeOfDay UTC:  00:00
   TimeZone:            Local
  IpSec Condition Summary:                NegativeIndicator: Off
   LocalDynVpn Condition:
    LocalIp:
     FromAddr:          9.3.3.3
     ToAddr:            9.3.3.3
    RemoteIp:
     FromAddr:          9.5.0.0
     Prefix:            16
    LocalDataPort:      12000             RemoteDataPort:    12000
    AutoActivate:       Yes
    Protocol:           UDP  (17)
  Policy created: Fri Sep 18 13:29:53 2011
  Policy updated: Fri Sep 18 13:29:53 2011
```

The following example shows all active IPSec policies names:

```
========================================================================
================== pasearch -v a -n ====================================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS1
  Date:                 09/18/2011        Time:  13:30:32
  IPSec Instance Id:    1253294993

policyRule:             Rule1Admin
  IpFilter Action:      permit

policyRule:             Rule2Admin
  IpFilter Action:      ipsec
  IpFilter Action:      Silver-TransportMode

policyRule:             Rule1A
  IpFilter Action:      permit

policyRule:             Rule2A
  IpFilter Action:      ipsec
  IpFilter Action:      Bronze-TransportMode

policyRule:             Rule1B
  IpFilter Action:      permit

policyRule:             Rule2B
  IpFilter Action:      ipsec
  IpFilter Action:      Gold-TransportMode

policyRule:             Rule1C
  IpFilter Action:      permit

policyRule:             Rule2C
  IpFilter Action:      ipsec
  IpFilter Action:      Gold-TunnelMode
  IpFilter Action:      StartZoneC

policyRule:             Rule1DtoC
  IpFilter Action:      permit

policyRule:             Rule2DtoC
  IpFilter Action:      ipsec
  IpFilter Action:      Gold-TunnelMode
  IpFilter Action:      StartZoneDtoZoneC

policyRule:             Rule1N
  IpFilter Action:      permit

policyRule:             Rule2N
  IpFilter Action:      ipsec
  IpFilter Action:      Gold-TransportMode

policyRule:             Rule1All-IPv4-Permit
  IpFilter Action:      permit

policyRule:             Rule2All-IPv4-Deny
  IpFilter Action:      deny-log

policyRule:             Rule1All-IPv6-Permit
  IpFilter Action:      permit

policyRule:             Rule2All-IPv6-Deny
  IpFilter Action:      deny-log

policyRule:             DenyAllRule_Generated___________Inbnd

policyRule:             DenyAllRule_Generated___________Outbnd
```

```
policyRule:             Admin_KeyExRule1
  KeyExchange Action:   Bronze-PSK

policyRule:             ZoneA_KeyExRule1
  KeyExchange Action:   Silver-RSA

policyRule:             ZoneB_KeyExRule1
  KeyExchange Action:   Gold-RSA

policyRule:             ZoneC_KeyExRule1
  KeyExchange Action:   Gold-RSA

policyRule:             ZoneN_KeyExRule1
  KeyExchange Action:   Gold-RSA-AllowNat

policyRule:             ZoneC_VPN-EE1

policyRule:             ZoneC_VPN-EE2

policyRule:             ZoneC_VPN-EE3

policyRule:             ZoneC_VPN-EE4

policyRule:             ZoneC_VPN-EE5

policyRule:             ZoneC_VPN-FTP-Data

policyRule:             ZoneC_VPN-FTP-Control

policyRule:             ZoneC_VPN-CICS-3000
```

The following example shows active IPFilter policies with Policy Action scope of DynamicVpn.

```
========================================================================
================== pasearch -s DynamicVpn -v f =========================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS1
  Date:                 09/18/2011        Time:  13:30:32
  IPSec Instance Id:    1253294993

policyRule:             Rule2Admin
  Rule Type:            IpFilter
  Version:              3                 Status:            Active
  GroupName:            Admin
  Weight:               119               ForLoadDist:       False
  Priority:             19                Sequence Actions:  Don't Care
  No. Policy Action:    2                 ConditionListType: CNF
  IpSecType:            policyIpFilter
  policyAction:         ipsec
   ActionType:          IpFilter GenericFilter
   Action Sequence:     0
  policyAction:         Silver-TransportMode
   ActionType:          IpFilter DynamicVpn
   Action Sequence:     0
  Time Periods:
   Day of Month Mask:
   First to Last:       1111111111111111111111111111111
   Last to First:       1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    1111111  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00             To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00             To TimeOfDay UTC:  04:00
```

```
 TimeZone:            Local
IpSec Condition Summary:                NegativeIndicator: Off
 IpFilter Condition:
  Source Address:
  Destination Address:
  Service Condition:
   Protocol:          0
   Direction:         0
   RouteType:         0                 SecurityClass:     0
   FragmentsOnly:     No
Condition Work Level:      0
  Group Number:       0                 Cond Count:        2
  Ignore:             No
IpSec Condition Work Summary:           NegativeIndicator: Off
 IpFilter Condition:
  Source Address:
  Destination Address:
  Service Condition:
   Protocol:          0
   Direction:         0
   RouteType:         0                 SecurityClass:     0
   FragmentsOnly:     No
IpSec Condition Work:                   NegativeIndicator: Off
 IpFilter Condition:
  Source Address:
   FromAddr:          9.1.1.1
   ToAddr:            9.1.1.1
  Destination Address:
  Service Condition:
   Protocol:          0
   Direction:         0
   RouteType:         0                 SecurityClass:     0
   FragmentsOnly:     No
Condition Work Level:      1
  Group Number:       1                 Cond Count:        2
  Ignore:             No
IpSec Condition Work Summary:           NegativeIndicator: Off
 IpFilter Condition:
  Source Address:
  Destination Address:
  Service Condition:
   Protocol:          0
   Direction:         0
   RouteType:         0                 SecurityClass:     0
   FragmentsOnly:     No
IpSec Condition Work:                   NegativeIndicator: Off
 IpFilter Condition:
  Source Address:
  Destination Address:
   FromAddr:          9.1.1.2
   ToAddr:            9.1.1.2
  Service Condition:
   Protocol:          0
   Direction:         0
   RouteType:         0                 SecurityClass:     0
   FragmentsOnly:     No
Condition Work Level:      2
  Group Number:       3                 Cond Count:        2
  Ignore:             No
IpSec Condition Work Summary:           NegativeIndicator: Off
 IpFilter Condition:
  Source Address:
  Destination Address:
  Service Condition:
   Protocol:          0
   Direction:         0
   RouteType:         0                 SecurityClass:     0
```

```
     FragmentsOnly:     No
  IpSec Condition Work:                   NegativeIndicator: Off
   IpFilter Condition:
    Source Address:
    Destination Address:
    Service Condition:
     Protocol:          All
     Direction:         Bidirectional
     RouteType:         Local             SecurityClass:     0
     FragmentsOnly:     No
  Policy created: Fri Sep 18 13:29:53 2011
  Policy updated: Fri Sep 18 13:29:53 2011

  IpFilter Action:      ipsec
    Version:            3                 Status:            Active
    Scope:              GenericFilter
    ipFilterAction:     IPSec             IpFilterLogging:   Yes Logdeny
    DiscardAction:      Silent
    Policy created: Fri Sep 18 13:29:53 2011
    Policy updated: Fri Sep 18 13:29:53 2011

  IpFilter Action:      Silver-TransportMode
    Version:            3                 Status:            Active
    Scope:              DynamicVpn
    Initiation:         Either            VpnLife:           1440
    AcceptablePfs:      None
    InitiateWithPfs:    None              IpDataOfferNum:    1
    PassthroughDSCP:    Yes               PassthroughDF:     Yes
    HowToEncapIKEv2:    Either
    IPDataOffer:        0
     HowToEncap:        Transport
     HowToEncrypt:      DES               KeyLength:         N/A
     HowToAuth:         ESP               HowToAuthAlgr:     HMAC_SHA1
     RefLifeTmPropose:  240
     RefLifeTmAcptMin:  120               RefLifeTmAcptMax:  480
     RefLifeSzPropose:  None
     RefLifeSzAccept :  None
    Policy created: Fri Sep 18 13:29:53 2011
    Policy updated: Fri Sep 18 13:29:53 2011
```

The following example shows active IDS policies whose names match the prefix AttackMalformed:

```
========================================================================
================== pasearch -i -w -f AttackMalformed ===================
========================================================================

TCP/IP pasearch CS V2R1                 Image Name: TCPCS2
  Date:                 09/28/2011        Time:  12:01:32
  IDS Instance Id:      1285689675

policyRule:             AttackMalformed-rule
  Rule Type:            IDS
  Version:              4                 Status:            Active
  Weight:               102               ForLoadDist:       False
  Priority:             2                 Sequence Actions:  Don't Care
  No. Policy Action:    1
  IdsType:              policyIdsAttack
  policyAction:         Attack-action
   ActionType:          IDS
   Action Sequence:     0
  Time Periods:
   Day of Month Mask:
   First to Last:       1111111111111111111111111111111
   Last to First:       1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    1111111  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00            To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00            To TimeOfDay UTC:  04:00
   TimeZone:            Local
  Ids Condition Summary:                 NegativeIndicator: Off
   Attack Condition:
    IdsAttackType:      MALFORMED_PACKET
  Policy created: Tue Sep 28 12:01:15 2011
  Policy updated: Tue Sep 28 12:01:15 2011

  Ids Action:             Attack-action
    Version:              4               Status:            Active
    Attack ActionType:    NoDiscard
    TypeActions:          Statistics Log
    StatType:             Exception       StatInterval:      60
    LogDetail:            No              LoggingLevel:      1
    Policy created: Tue Sep 28 12:01:15 2011
    Policy updated: Tue Sep 28 12:01:15 2011
```

The following example shows active IDS rules and actions configured from the IDS configuration file:

```
========================================================================
================== pasearch -i  ========================================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS2
  Date:                 09/28/2011       Time:  12:01:55
  IDS Instance Id:      1285689675

policyRule:             ScanEventLowTcp-rule
  Rule Type:            IDS
  Version:              4                Status:            Active
  Weight:               102              ForLoadDist:       False
  Priority:             2                Sequence Actions:  Don't Care
  No. Policy Action:    1
  IdsType:              policyIdsScanEvent
  policyAction:         ScanEventLow-action
   ActionType:          IDS
   Action Sequence:     0
  Time Periods:
   Day of Month Mask:
   First to Last:       1111111111111111111111111111111
   Last to First:       1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    1111111  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00            To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00            To TimeOfDay UTC:  04:00
   TimeZone:            Local
  Ids Condition Summary:                 NegativeIndicator: Off
   ScanEvent Condition:
    Sensitivity:        Low
    Protocol:           TCP  (6)
    LocalPortFrom:      1                LocalPortTo:       1023
    LocalHostAddress:
     FromAddr:          All
     ToAddr:            All
  Policy created: Tue Sep 28 12:01:15 2011
  Policy updated: Tue Sep 28 12:01:15 2011

  Ids Action:             ScanEventLow-action
    Version:              4               Status:            Active
    ScanEvent ActionType: Count
    Policy created: Tue Sep 28 12:01:15 2011
    Policy updated: Tue Sep 28 12:01:15 2011
```

The following example shows active AT-TLS policies:

```
========================================================================
================== pasearch -t  ========================================
========================================================================

policyRule:             Secure_Telnet_23_Debug
  Rule Type:            TTLS
  Version:              3                 Status:            Active
  Weight:               20                ForLoadDist:       False
  Priority:             20                Sequence Actions:  Don't Care
  No. Policy Action:    3
  policyAction:         grp_Production
   ActionType:          TTLS Group
   Action Sequence:     0
  policyAction:         Secure_Telnet_Env
   ActionType:          TTLS Environment
   Action Sequence:     0
  policyAction:         Secure_Telnet_Conn_Debug
   ActionType:          TTLS Connection
   Action Sequence:     0
  Time Periods:
   Day of Month Mask:
   First to Last:       1111111111111111111111111111111
   Last to First:       1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    1111111  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        00:00             To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00             To TimeOfDay UTC:  04:00
   TimeZone:            Local
  TTLS Condition Summary:                 NegativeIndicator: Off
   Local Address:
    FromAddr:           10.1.2.3
    ToAddr:             10.1.2.3
   Remote Address:
    FromAddr:           10.45.23.10
    ToAddr:             10.45.23.10
   LocalPortFrom:       23                LocalPortTo:       23
   RemotePortFrom:      0                 RemotePortTo:      0
   JobName:                               UserId:
   ServiceDirection:    Inbound
  Policy created: Wed Mar  9 06:31:13 2011
  Policy updated: Wed Mar  9 06:31:13 2011

  TTLS Action:                  grp_Production
    Version:                    3
    Status:                     Active
    Scope:                      Group
    TTLSEnabled:                On
    CtraceClearText:            Off
    Trace:                      2
    FIPS140:                    Off
    TTLSGroupAdvancedParms:
     SecondaryMap:              Off
     SyslogFacility:            Daemon
    Policy created: Wed Mar  9 06:31:13 2011
    Policy updated: Wed Mar  9 06:31:13 2011

  TTLS Action:                  Secure_Telnet_Env
    Version:                    3
    Status:                     Active
    Scope:                      Environment
    HandshakeRole:              Server
    SuiteBProfile:              Off
    TTLSKeyringParms:
     Keyring:                   TCPCSsafkeyring
    TTLSEnvironmentAdvancedParms:
     SSLv2:                     Off
     SSLv3:                     On
     TLSv1:                     On
     TLSv1.1:                   On
     TLSv1.2:                   Off
     ApplicationControlled:     On
     HandshakeTimeout:          5
     ClientAuthType:            Required
     ResetCipherTimer:          0
     TruncatedHMAC:             Off
     CertValidationMode:        Any
     ServerMaxSSLFragment:      Off
     ClientMaxSSLFragment:      Off
     ServerHandshakeSNI:        Off
     ClientHandshakeSNI:        Off
     Renegotiation:             Default
     RenegotiationIndicator:    Optional
     RenegotiationCertCheck:    Off
    EnvironmentUserInstance:    0
    Policy created: Wed Mar  9 06:31:13 2011
    Policy updated: Wed Mar  9 06:31:13 2011

  TTLS Action:                  Secure_Telnet_Conn_Debug
    Version:                    3
    Status:                     Active
    Scope:                      Connection
    CtraceClearText:            On
    Trace:                      254
    Policy created: Wed Mar  9 06:31:13 2011
    Policy updated: Wed Mar  9 06:31:13 2011
```

The following example shows active routing policies:

```
========================================================================
================== pasearch -R  ========================================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS3
  Date:                 10/12/2012        Time:  11:00:46
  Routing Instance Id:  1350050178

policyRule:             GenericRoutingRule
  Rule Type:            Routing
  Version:              4                 Status:            Active
  Weight:               10                ForLoadDist:       False
  Priority:             10                Sequence Actions:  Don't Care
  No. Policy Action:    1
  policyAction:         GenericRoutingAction
   ActionType:          Routing
   Action Sequence:     0
  Time Periods:
   Day of Month Mask:
   First to Last:       1111111111111111111111111111111
   Last to First:       1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    1111111  (Sunday - Saturday)
   Start Date Time:     None
   End Date Time:       None
   Fr TimeOfDay:        08:00             To TimeOfDay:      17:00
   Fr TimeOfDay UTC:    11:00             To TimeOfDay UTC:  20:00
   TimeZone:            Local
  Routing Condition Summary:              NegativeIndicator: Off
   IpSourceAddr Address:
    FromAddr:           All
    ToAddr:             All
   IpDestAddr Address:
    FromAddr:           0.0.0.0
    Prefix:             0
   TrafficDescriptor:
    Protocol:           TCP  (6)
    SourcePortFrom      111               SourcePortTo       111
    DestinationPortFrom 1024              DestinationPortTo  65535
    JobName             JOB1              SecurityZone       SECZONE
    SecurityLabel       SECLABEL
  Policy created: Fri Oct 12 10:56:18 2012
  Policy updated: Fri Oct 12 10:56:18 2012

  Routing Action:       GenericRoutingAction
    Version:            4                 Status:            Active
    UseMainRouteTable   Yes
    RouteTable:         RtTbl1
    RouteTable:         RtTbl2
    RouteTable:         RtTbl3
    Policy created: Fri Oct 12 10:56:18 2012
    Policy updated: Fri Oct 12 10:56:18 2012
```

The following example shows active route tables:

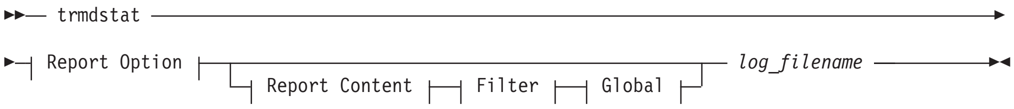
```
========================================================================
================== pasearch -T  ========================================
========================================================================

TCP/IP pasearch CS V2R1                  Image Name: TCPCS3
  Date:                 10/12/2012        Time:  11:03:00
  Routing Instance Id:  1350050178

  Route Table:           RtTbl1
    Version:             1                 Status:            Active
    IPv4 table           Active
    IgnorePathMtuUpdate  No
    MultiPath            PerConnection     DynamicXCFRoutes   No
    IPv6 table           Active
    IgnorePathMtuUpdate6 No
    MultiPath6           PerConnection     DynamicXCFRoutes6  No
    Route (IPv4)
     Destination:
      ipaddress            1.1.1.1
     First Hop:
      gateway_addr         =
      link_name            LINK1
     MTU size             1492
     Replaceable          No
     MaximumRetransmitTime 120.000
     MinimumRetransmitTime 0.500
     RoundTripGain        0.125
     VarianceGain         0.250
     VarianceMultiplier   2.000
     DelayAcks            Yes
    Route (IPv4)
     Destination:
      ipaddress            1.0.0.0
      Prefix               8
     First Hop:
      gateway_addr         2.2.2.2
      link_name            LINK2
     MTU size             1492
     Replaceable          No
     MaximumRetransmitTime 120.000
```

```
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Route (IPv4)
   Destination           Default
   First Hop:
    gateway_addr         4.4.4.4
    link_name            LINK4
   MTU size              DEFAULTSIZE
   Replaceable           No
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Route (IPv6)
   Destination:
    ipaddress            2001:db8:0:0:1::
    Prefix               80
   First Hop:
    gateway_addr         fe80::2:2:2:2
    link_name            LINK2V6
   MTU size              5000
   Replaceable           No
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Route (IPv6)
   Destination           Default6
   First Hop:
    gateway_addr         fe80::4:4:4:4
    link_name            LINK4V6
   MTU size              DEFAULTSIZE
   Replaceable           No
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Policy created: Fri Oct 12 10:56:18 2012
  Policy updated: Fri Oct 12 10:56:18 2012

Route Table:           RtTbl2
  Version:             1                 Status:            Active
  IPv4 table           Active
  IgnorePathMtuUpdate  No
  MultiPath            UseGlobal         DynamicXCFRoutes   No
  IPv6 table           Active
  IgnorePathMtuUpdate6 No
  MultiPath6           UseGlobal         DynamicXCFRoutes6  No
  DynamicRoutingParms (IPv4)
   link_name             LINK1     IPv4
  DynamicRoutingParms (IPv4)
   link_name             LINK2
   gateway_addr          2.1.1.1
  DynamicRoutingParms (IPv4)
   link_name             LINK2
   gateway_addr          2.2.2.2
  DynamicRoutingParms (IPv6)
   link_name             LINK1V6     IPv6
  DynamicRoutingParms (IPv6)
   link_name             LINK2V6
   gateway_addr          fe80::2:1:1:1
  Policy created: Fri Oct 12 10:56:18 2012
  Policy updated: Fri Oct 12 10:56:18 2012

Route Table:           RtTbl3
  Version:             1                 Status:            Active
  IPv4 table           Active
  IgnorePathMtuUpdate  No
  MultiPath            UseGlobal         DynamicXCFRoutes   No
  IPv6 table           Active
  IgnorePathMtuUpdate6 No
  MultiPath6           UseGlobal         DynamicXCFRoutes6  No
  Route (IPv4)
   Destination:
    ipaddress            1.1.1.1
   First Hop:
    gateway_addr         =
    link_name            LINK1
   MTU size              1492
   Replaceable           No
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Route (IPv4)
   Destination:
    ipaddress            1.1.0.0
    Prefix               16
   First Hop:
    gateway_addr         2.2.2.2
    link_name            LINK2
   MTU size              1492
   Replaceable           Yes
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Route (IPv6)
   Destination:
    ipaddress            2001:db8::1:1:0:0
    Prefix               96
   First Hop:
    gateway_addr         fe80::2:2:2:2
    link_name            LINK2V6
   MTU size              5000
   Replaceable           Yes
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  DynamicRoutingParms (IPv4)
   link_name             LINK2     IPv4
  DynamicRoutingParms (IPv6)
   link_name             LINK2V6     IPv6
  Policy created: Fri Oct 12 10:56:18 2012
  Policy updated: Fri Oct 12 10:56:18 2012
```

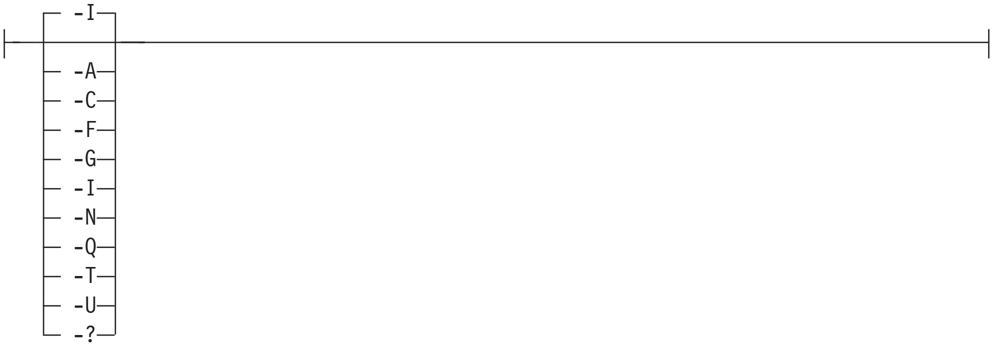
The z/OS UNIX trmdstat command: Display traffic regulation management daemon (TRMD) log

Use the **trmdstat** command to give a consolidated view of the IDS log messages written out by the Traffic Regulation Management daemon (TRMD).

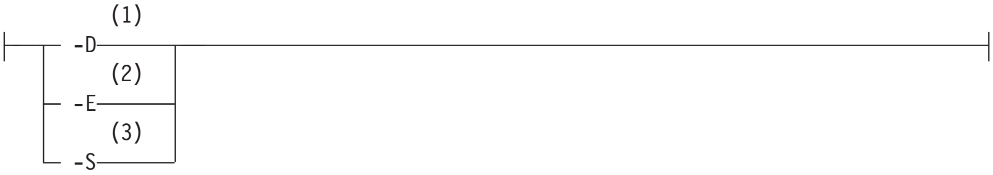
z/OS UNIX trmdstat command syntax



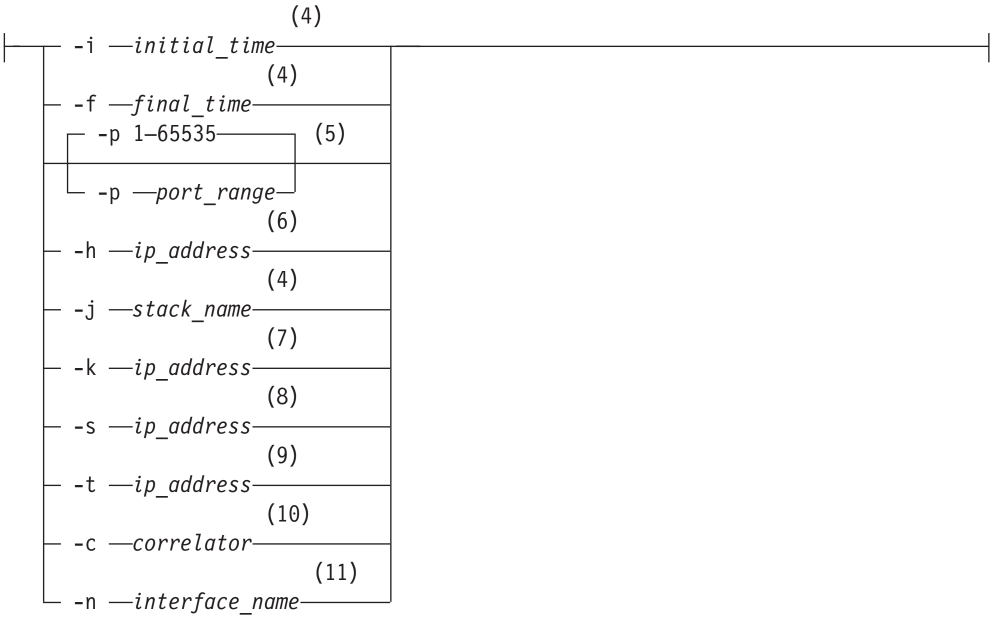
Report Option:



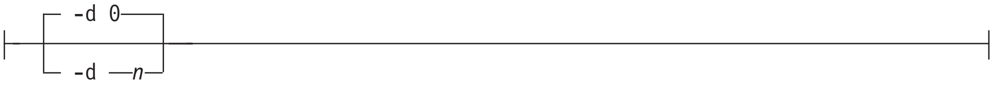
Report Content:



Filter:



Global:



Notes:

1 Valid only when -A/-C/-F/-G/-N/-Q/-T/-U is specified.

2 Valid only when -T is specified.

3 Valid only when -A/-F/-T/-U is specified.

4 Valid only when -A/-C/-F/-G/-I/-N/-Q/-T/-U is specified.

5 Valid only when -A/-C/-F/-G/-Q/-T/-U is specified except when -A -S or -F -S are specified.

6 Valid only when -A/-C/-F/-G/-N/-Q/-U is specified except when -A -S is specified.

7 Valid only when -T and -S is specified.

8 Valid only when -A/-G/-Q/-T is specified except when -A -S or -T -S are specified.

9 Valid only when -A/-G/-Q/-T is specified except when -A -S is specified.

10 Not valid when -S or -I is specified.

11 Valid only when -F is specified.

z/OS UNIX trmdstat command parameter descriptions

The following topics describe the individual parameter items contained in the syntax diagram.

z/OS UNIX trmdstat command report options

The following report options can be used with the trmdstat command. If no report option is specified, trmdstat displays the default -I report.

-A Displays the attack summary.

-C Displays the connection summary.

-F Displays the flood summary.

-G Displays the Global TCP Stall summary.

-I Displays the IDS Overall Summary Report.

-N Displays the scan summary.

-Q Displays the TCP Queue Size summary

-T Displays the TCP TR summary.

-U Displays the UDP TR summary.

log_file_name
Name of the input file to be analyzed (the logfile of TRMD). You must enter a *log_file_name*.

-?
Displays the help information.

z/OS UNIX trmdstat command report content options

The following parameters can be used to specify the level of detail of the report content.

-D Displays detailed information. Valid only when the **-A/-C/-F/-G/-N/-Q/-T/-U** option is used (for example, **-AD** or **-A -D**).

-E Specifies the TCP extended summary report. Valid only with **-T**.

-S Displays statistics summary. Valid only when **-A/-F/-T/-U** is specified.

z/OS UNIX trmdstat command filter options

The following parameters can be used to filter the output of the specified report.

-i *initial_time*
The time of the first record to be considered. If this option is not specified, the first available record in the file is selected. The time is specified in the format `MMDDHHMMSS`.

MM	Month
DD	Date
HH	Hours
MM	Minutes
SS	Seconds

For example, `1021143030` is Oct 21 14:30:30. Trailing zeros are not required (`1021` for Oct 21 00:00:00).

For records generated by the TCP stack, the time the event actually occurred (the stack time) is used for the time filtering.

TRMD can also write syslog messages, for example, the EZZ8495I TRMD STARTED and the EZZ8501I TRMD ENDED messages. These messages contain only the syslog timestamp, which is used to filter these messages. The offset from the Coordinated Universal Time (UTC) of the syslog time is determined by the TZ environment variable when TRMD is started. For more information about setting the UTC offset, see the TRMD section in Intrusion Detection Services inz/OS Communications Server: IP Configuration Guide.

-f *final_time*
The time of the last record to be considered. If this option is not specified, the last record time available in the file is used. The format of the time is the same as in *initial_time*.

-p *port_range*
The port range to be considered. If this is not specified, all the ports are considered. The *port_range* value can be specified as follows:

- A single port: `-p  21`
- A range of ports: `-p 21-220`

Valid only when -A/-C/-F/-G/-Q/-T/-U option is used except when the -A -S, or -F -S options are specified.

- For the attack summary (-A) and attack detail (-AD), the port_range filter value will be matched to the destination port in the messages.
- For the connection summary (-C) and connection detail (-CD), the port_range filter value will be matched to the local port in the messages.
- For the flood summary (-F) and flood detail (-FD), the port_range filter value will be matched to the bound port in the SYN flood messages.
- For the global TCP stall summary (-G) and global TCP stall detail (-GD), the port_range filter will be matched to the local port in the messages.
- For the TCP queue size summary (-Q) and TCP queue size detail (-QD), the port_range filter will be matched to the local port in the messages.
- For the TCP TR summary (-T), TCP TR extended summary (-TE), TCP TR detail (-TD), and TCP TR statistic (-TS), the port_range filter value will be matched to the local port in the messages.
- For the UDP TR summary (-U), UDP TR detail (-UD), and UDP TR statistic (-US), the port_range filter value will be matched to the local port in the messages.

-h *ip_address*
Displays information about that particular IP address. Valid only when the -A/-C/-F/-G/-N/-Q/-U option is used except when the -A -S options are specified.

- For the attack summary (-A) and attack detail (-AD), the ip_address filter value will be matched to the destination address in the messages.
- For the connection summary (-C) and connection detail (-CD), the ip_address filter value will be matched to the source address in the messages.
- For the flood summary (-F) and flood detail (-FD), the ip_address filter value will be matched to the bound address in the SYN flood messages, the destination address in the Interface flood messages, and the destination address in the EE XID flood messages.
- For the global TCP stall summary (-G) and global TCP stall detail (-GD), the ip_address filter value will be matched to the remote host address in the messages.
- For the scan summary (-N) and scan detail (-ND) reports, the ip_address filter value will be matched to the source address in the messages.
- For the TCP queue size summary (-Q) and TCP queue size detail (-QD), the ip_address filter value will be matched to the remote host address in the messages.

- For the UDP TR summary (-U), UDP TR detail (-UD), and UDP TR statistic (-US), the ip_address filter value will be matched to the local IP address in the messages.

-j *stack_name*
Only messages containing the specified stack name are included in the report. The stack name is limited to eight characters.

-k *ip_address*
Specifies that information is to be gathered about the peak *ip_address*. Valid only when the **-T** and **-S** options are specified together.

-s *ip_address*
Specifies that information is to be gathered about the source *ip_address*. Valid only when the -A/-G/-Q/-T option is used except when the -A -S, or -T -S options are specified.

- For the attack summary (-A) and attack detail (-AD), the ip_address filter value will be matched to the source address in the messages.
- For the global TCP stall summary (-G) and global TCP stall detail (-GD), the ip_address filter value will be matched to the remote host address in the messages.
- For the TCP queue size summary (-Q) and TCP queue size detail (-QD), the ip_address filter value will be matched to the remote host address in the messages.
- For the TCP TR summary (-T), TCP TR extended summary (-TE), and TCP TR detail (-TD), the ip_address filter value will be matched to the source host address in the messages.

-t *ip_address*
Specifies that information is to be gathered about the destination *ip_address*. Valid only when the -A/-G/-Q/-T option is used except when the -A -S option is specified.

- For the attack summary (-A) and attack detail (-AD), the ip_address filter value will be matched to the destination address in the messages.
- For the global TCP stall summary (-G) and global TCP stall detail (-GD), the ip_address filter value will be matched to the local host address in the messages.
- For the TCP queue size summary (-Q) and TCP queue size detail (-QD), the ip_address filter value will be matched to the local host address in the messages.
- For the TCP TR summary (-T), TCP TR extended summary (-TE), TCP TR detail (-TD), and TCP TR statistic (-TS), the ip_address filter value will be matched to the local host address in the messages.

-c *correlator*
Specifies that information is to be gathered for records with the specified correlator. Not valid with **-S** or **-I**.

-n *interface_name*
Specifies that information is to be gathered about the interface (or Link). Valid only when **-F** is specified. If interface name is not applicable, such as in overall flood data, the record is not selected. The interface name is case sensitive and must be specified as shown in the report.

z/OS UNIX trmdstat command global options

The following parameter can be used to get debug information with any report option.

-d *n*

Specifies the debug level. The default level is 0, no debug. The higher the debug level, the greater the number of messages that are displayed. The valid debug levels are in the range 0 - 2.

z/OS UNIX trmdstat command report details and examples

This section contains descriptive information about the formatting and contents of trmdstat reports, including examples.

The trmdstat report general concept

To fully understand the following concepts and fields, you need to have some general knowledge of intrusion detection services (IDS). See Intrusion Detection Services in z/OS Communications Server: IP Configuration Guide for more information about Intrusion Detection Services.

The trmdstat command report heading:

All display reports from the trmdstat command begin with heading lines, which give general information related to the request.

```
trmdstat for z/OS CS V2R1        Wed Feb 23 07:53:11 2011

Command Entered      : trmdstat -A /tmp/syslog.log
Log Time Interval    : Feb 22 12:32:49  - Feb 22 16:19:53
Stack Time Interval  : Feb 22 17:32:19  - Feb 22 21:19:35
TRM Records Scanned  : 336
```

The heading lines contain the following fields:

CommandName
The command name.

VersionRelease
The version and release of trmdstat that was used to generate the report.

timestamp
The date and time the report was generated.

Command Entered
The command that was entered, including all specified parameters and the name of the input file.

Log Time Interval
The time interval from which syslog records were processed to generate the report, based on the syslog timestamp in the records. The syslog timestamp represents the time that the record was written to syslogd. This timestamp is dependent on the setting of the TZ environment variable at the time when TRMD is started. See the TRMD section in Intrusion Detection Services inz/OS Communications Server: IP Configuration Guide for more information about setting the TZ variable.

Stack Time Interval
The time interval from which syslog records were processed to generate the report, based on the stack timestamp in the records. The stack timestamp represents the time that the event was detected by the stack. This timestamp is always Coordinated Universal Time (UTC).

TRM Records Scanned
The number of records in the input file that were scanned while generating the report.

Messages lost:

This data comes from an EZZ9325I or EZZ9326I message. An EZZ9325I or EZZ9326I message is written to syslog when TRMD is unable to write syslog messages rapidly enough to keep up with the stack, and the storage allocated to contain messages is overwritten with new ones before the old ones are copied to syslog by TRMD. If this event occurs, consider increasing the priority of the TRMD task or reducing the amount of logging activity by changing IDS policy. For example,

```
07/09/2011 14:34:05.02 Number of ATTACK messages lost: 10

07/30/2011 01:34:05.05 Number of TCPTR  messages lost: 22
```

Messages-lost data can appear at the end of a trmdstat report. These messages do not contain detailed information and so are not included in the trmdstat report totals. However, if messages that relate to the requested trmdstat report exist, information from these messages is written at the end of the report.

Log-suppressed messages:

Log-suppressed messages can appear at the end of a trmdstat report. These messages can occur if log messages were suppressed by IDS to prevent possible flooding of syslog. Both TCP traffic regulation and IDS attack detection limit the number of log records that can be written in a 5-minute interval. If the limit is exceeded, the log record is not written. However, at the end of the 5-minute interval, a log record is written that indicates the number of suppressed log records. EZZ8660I, EZZ8661I, and EZZ9327I are log-suppressed messages. These messages do not contain detail information and, therefore, are not included in the trmdstat report totals. However, if messages that relate to the requested trmdstat report exist, information from these messages is written at the end of the report.

TCP TR message suppression

TCP TR limits the number of connection refused (EZZ9324I), would have been refused (EZZ9319I), QoS exception made (EZZ9317I), or QoS exception logged (EZZ9318I) records written in a 5-minute interval. For a listening port, a maximum of 100 of these log records is written within a 5-minute interval. Globally, TCP TR writes a maximum of 1000 log records in a 5-minute interval. If a log record was not written because of these limits, the count of refused or would have been refused connections log records that were not logged is recorded in the EZZ8660I TRMD TCP connection log records suppressed log record after the 5-minute interval ends. Similarly, the count of QoS exception records that were not written is recorded in the EZZ8661I TRMD TCP QOS exception log records suppressed log message. The counts from these messages are not included in the trmdstat report totals. Instead, the counts are listed at the end of the requested report in the following format:

datetime Number of TCP *type* messages suppressed for local host *laddr* port *lport* scope *rsn: count*

For example:

```
07/30/2011 01:35:05.01 Number of TCP connection refused messages suppressed for local host :: port 345 scope TR: 8
07/30/2011 01:35:05.01 Number of TCP QOS exception messages suppressed for local host 50c9:c2d4::9:42:105:25 port 25 scope Port: 6
```

The values in the report follow:

datetime
The stack date and time when the first log record in the 5-minute interval was suppressed.

type The type of suppressed TCP connections. Can be connection refused or QOS exception.

laddr The local IP address.

lport The local listening port.

rsn The reason that the log records were suppressed. Can be one of the following values:

Port The log record was suppressed because 100 log records had already been written for the listening port in the 5-minute interval.

TR The log record was suppressed because the total number of TCP TR log records written during the five minute interval exceeded 1000 log records.

count The number of log records suppressed during the 5-minute interval.

The TCP TR suppressed count messages can be written for the IDS Overall Summary report and any of the TCP summary or detail reports that are requested with the -T or -C options.

If the trmdstat report requested filtering by source IP address (with either the -h or -s options), the suppressed count messages are not included following the report because the source IP address is not included in the TCP suppressed messages (EZZ8660I and EZZ8661I). However, if there were suppressed messages that met all the other filtering criteria, the following warning message is written at the end of the report:`Suppressed messages do not contain filter information and are not displayed`.

If this message occurs and the suppressed count messages are required, reissue the trmdstat request without the -h or -s options or request the IDS Overall Summary report (-I option).

Attack message suppression

IDS attack-processing limits the number of log records written for a particular attack type to 100 records in a 5-minute interval. The following messages can be suppressed: EZZ8648I, EZZ8649I, EZZ8662I, EZZ8663I, EZZ8664I, EZZ8665I, EZZ8666I, EZZ8667I, EZZ8668I, EZZ8669I, EZZ8670I, EZZ8671I, EZZ8672I, EZZ8675I, EZZ8676I, EZZ8677I, EZZ8678I. If a log record was not written because of this limit, the number of suppressed messages is recorded in the EZZ9327I Attack log records suppressed message. The counts from these messages are not included in the trmdstat report totals. The counts are listed at the end of the requested report in the following format:

datetime Number of ATTACK *type* messages suppressed: *count*

Examples:

```
07/09/2011 15:53:03.94 Number of ATTACK Malform  messages suppressed: 3
07/09/2011 15:53:05.94 Number of ATTACK OutRaw4  messages suppressed: 4
07/09/2011 15:53:06.94 Number of ATTACK PerpEcho messages suppressed: 3
```

The values are:

datetime
The stack date and time when the first log record in the 5-minute interval was suppressed.

type The attack type. It can be one of the following types: DataHide, DestOpts, EELDLC, EEMalfmd, EEPort, EEXID, Flood, Fragment, HopOpts, IPOption, IPProto, Malform, NextHdrs, OutRaw4, OutRaw6, PerpEcho, Redirect, TCPQueSz, TCPStall, NoId.

count The number of log records suppressed during the 5-minute interval.

The attack-suppressed count messages can be written for the IDS overall summary report and any of the attack summary or detail reports.

Attack summary (-A) report

This report is displayed when the -A option is specified with the trmdstat command. It displays the summary of all attack events. The information presented in this report is derived from EZZ8648I and EZZ8649I types of syslog messages. Information is grouped by destination IP address - source IP address pair. It is sorted by destination IP address and then by destination port.

```
>trmdstat -A /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:12:26 2011

Command Entered     : trmdstat -A /tmp/tstlog.log
Log Time Interval   : Nov 12 04:36:51  - Nov 29 19:55:50
Stack Time Interval : Nov 12 04:36:47  - Nov 29 19:55:46
TRM Records Scanned : 227

                            ATTACK Summary

                            Packets Discarded

Destination IP Address: 192.168.105.53
Source IP Address:      192.168.105.50

Dest  Malform/   OutRaw4/   Redirect/  DestOpts/  IPProto/   PerpEcho/  EELDLC/
Port  Fragment   OutRaw6    IPOption   HopOpts    NextHdrs   DataHide   EEPort     EEMalfmd   NoId
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
11000          0          0          0          0          0          0          0          1          0
               0          0          0          0          0          0          0
12000          0          0          0          0          0          0          0          2          0
               0          0          0          0          0          0          1

                            Packets Discarded

Destination IP Address: 2001:db8:0:3:9:42:103:132
Source IP Address:      2001:db8::20d:60ff:fe24:32ae

Dest  Malform/   OutRaw4/   Redirect/  DestOpts/  IPProto/   PerpEcho/  EELDLC/
Port  Fragment   OutRaw6    IPOption   HopOpts    NextHdrs   DataHide   EEPort     EEMalfmd   NoId
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
    0          0          0          1          0          0          0          0          0          0
               0          1          0          0          0          0          0

                            Packets Discarded

Destination IP Address: 2001:db8:0:3:9:42:103:132
Source IP Address:      2001:db8:0:3:20a:5eff:fe04:8f16

Dest  Malform/   OutRaw4/   Redirect/  DestOpts/  IPProto/   PerpEcho/  EELDLC/
Port  Fragment   OutRaw6    IPOption   HopOpts    NextHdrs   DataHide   EEPort     EEMalfmd   NoId
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
    0          2          0          0          0          0          0          0          0          0
               0          0          0          0          2          0          0

                            Packets Would Have Been Discarded

Destination IP Address: 192.168.0.5
Source IP Address:      192.168.101.3

Dest  Malform/   OutRaw4/   Redirect/  DestOpts/  IPProto/   PerpEcho/  EELDLC/
Port  Fragment   OutRaw6    IPOption   HopOpts    NextHdrs   DataHide   EEPort     EEMalfmd   NoId
```

```
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
    0          0          1          0          0          0          0          0          0          0
               0          0          0          0          0          0          0

                            Packets Would Have Been Discarded

Destination IP Address: 2001:db8:0:3:9:42:103:132
Source IP Address:      2001:db8::20d:60ff:fe24:32ae

Dest  Malform/   OutRaw4/   Redirect/  DestOpts/  IPProto/   PerpEcho/  EELDLC/
Port  Fragment   OutRaw6    IPOption   HopOpts    NextHdrs   DataHide   EEPort     EEMalfmd   NoId
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
    0          0          0          1          0          0          0          0          0          0
               0          0          0          0          0          0          0
    7          0          0          0          0          0          1          0          0          0
               0          0          0          0          0          0          0

                            Packets Would Have Been Discarded

Destination IP Address: 2001:db8:0:3:9:42:103:132
Source IP Address:      2001:db8:0:3:20a:5eff:fe04:8f16

Dest  Malform/   OutRaw4/   Redirect/  DestOpts/  IPProto/   PerpEcho/  EELDLC/
Port  Fragment   OutRaw6    IPOption   HopOpts    NextHdrs   DataHide   EEPort     EEMalfmd   NoId
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
    0          0          0          0          2          0          0          0          0          0
               0          0          0          1          1          0          0
```

The following information describes the areas of the ATTACK summary report.

Destination IP Address
Specifies the destination IP address.

Source IP Address
Specifies the source IP address.

DestPort
Specifies the destination port number.

Malform
Specifies the number of malformed packet attacks detected.

Fragment
Specifies the number of IP fragment packet attacks detected.

OutRaw4
Specifies the number of outbound IPv4 raw packet attacks detected.

OutRaw6
Specifies the number of outbound IPv6 raw packet attacks detected.

Redirect
Specifies the number of ICMP Redirect packet attacks detected.

IPOption
Specifies the number of restricted IPv4 option packet attacks detected.

DestOpts
Specifies the number of restricted IPv6 destination option packet attacks detected.

HopOpts
Specifies the number of restricted IPv6 hop-by-hop option packet attacks detected.

IPProto
Specifies the number of restricted IPv4 protocol packet attacks detected.

NextHdrs
Specifies the number of restricted IPv6 next header packet attacks detected.

PerpEcho
: Specifies the number of perpetual echo packet attacks detected.

DataHide
: Specifies the number of packets detected with possible hidden data.

EELDLC
: Specifies the number of EE LDLC packets detected that were received on the wrong port.

EEPort
: Specifies the number of EE packets detected with the incorrect source port value.

EEMalfmd
: Specifies the number of EE malformed packets detected.

NoId
: Specifies the number of EZZ8648I or EZZ8649I messages received with an unknown attack type. It might be that the version of the z/OS Communications Server on which the trmdstat command is being run is older than the version of z/OS Communication Server that detected the attacks.

Packets Discarded
: A report section header indicating packets that were discarded.

Packets Would Have Been Discarded
: A report section header indicating packets that would have been discarded.

messages suppressed
: The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

Attack detail (-A -D) report

This report is displayed when both the -A and -D options are specified on the trmdstat command. It displays the contents of attack event records. The information presented in this report is derived from EZZ8648I and EZZ8649I types of syslog messages. Information is grouped by destination IP address - source IP address pair. It is sorted by destination IP address.

```
>trmdstat -AD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:13:17 2011

Command Entered     : trmdstat -AD /tmp/tstlog.log
Log Time Interval   : Nov 12 04:36:51  - Nov 29 19:55:50
Stack Time Interval : Nov 12 04:36:47  - Nov 29 19:55:46
TRM Records Scanned : 227

                                         ATTACK Events

                                         Packets Discarded

                                           Destination IP Address/            DestPort/
 Attack      Date and Time                   Source IP Address                 SrcPort   Correlator ProbeID
-------- ---------------------- --------------------------------------------- --------- ---------- --------
EEPortCk 11/12/2011 04:36:47.22 192.168.105.53                                    12000          4 04120001
                                192.168.105.50                                     8000

EEMalfmd 11/12/2011 04:38:54.39 192.168.105.53                                    12000          5 04110001
                                192.168.105.50                                    12000

EEMalfmd 11/12/2011 04:39:19.41 192.168.105.53                                    12000          6 04110001
                                192.168.105.50                                    12000

EEMalfmd 11/12/2011 04:39:41.59 192.168.105.53                                    11000          7 04110001
                                192.168.105.50                                    11000

Redirect 11/12/2011 18:52:43.38 2001:db8:0:3:9:42:103:132                             0         10 04040001
```

```
                                2001:db8::20d:60ff:fe24:32ae                          0

OutRaw6  11/29/2011 19:55:46.59 2001:db8:0:3:9:42:103:132                             0          70 040C0001
                                2001:db8::20d:60ff:fe24:32ae                          0

NextHdrs 11/18/2011 16:02:47.44 2001:db8:0:3:9:42:103:132                             0          31 040D0001
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0

NextHdrs 11/18/2011 16:02:53.51 2001:db8:0:3:9:42:103:132                             0          32 040D0001
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0

Malform  11/13/2011 15:06:51.12 2001:db8:0:3:9:42:103:132                             0          12 0401003D
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0

Malform  11/13/2011 15:06:56.80 2001:db8:0:3:9:42:103:132                             0          16 0401003D
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0


                                         Packets Would Have Been Discarded

                                           Destination IP Address/            DestPort/
 Attack      Date and Time                   Source IP Address                 SrcPort   Correlator ProbeID
-------- ---------------------- --------------------------------------------- ---------  ---------- --------
OutRaw4  11/29/2011 19:24:27.02 192.168.0.5                                           0          63 04020001
                                192.168.101.3                                         0

Redirect 11/12/2011 17:27:57.30 2001:db8:0:3:9:42:103:132                             0           9 04040001
                                2001:db8::20d:60ff:fe24:32ae                          0

PerpEcho 11/14/2011 15:53:16.14 2001:db8:0:3:9:42:103:132                             7          24 04080003
                                2001:db8::20d:60ff:fe24:32ae                          7

NextHdrs 11/19/2011 14:10:01.02 2001:db8:0:3:9:42:103:132                             0          38 040D0001
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0

DestOpts 11/13/2011 15:06:51.11 2001:db8:0:3:9:42:103:132                             0          11 040E0001
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0

DestOpts 11/13/2011 15:06:56.80 2001:db8:0:3:9:42:103:132                             0          15 040E0001
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0

HopOpts  11/13/2011 15:07:14.05 2001:db8:0:3:9:42:103:132                             0          17 040F0001
                                2001:db8:0:3:20a:5eff:fe04:8f16                       0
```

The following information describes the areas of the attack detail report.

Attack Specifies the attack type. The values that can be displayed are:

- DataHide - Data hiding
- DestOpts - Restricted IPv6 destination option
- EELDLCCk - Enterprise Extender LDLC check
- EEMalfmd - EE malformed packet
- EEPortCk - EE source port check
- Fragment - IP Fragment
- HopOpts - Restricted IPv6 hop-by-hop option
- IPOption - Restricted IPv4 option
- IPProto - Restricted IPv4 protocol
- Malform - Malformed packet
- NextHdrs - Restricted IPv6 next header
- OutRaw4 - Outbound IPv4 Raw
- OutRaw6 - Outbound IPv6 Raw
- PerpEcho - Perpetual echo
- Redirect- ICMP redirect

- Blank - The attack type is unrecognized. It might be that the version of the z/OS Communications Server running the trmdstat command is older than the version of z/OS Communication Server that detected the attack.

Date and Time
Specifies the date and time.

Destination IP Address
Specifies the destination IP address.

Source IP Address
Specifies the source IP address.

DestPort
Specifies the destination port.

SrcPort
Specifes the source port.

Correlator
Specifies the trace correlator.

ProbeID
Specifies the IDS probeID that generated this event.

Packets Discarded
A report section header indicating packets that were discarded.

Packets Would Have Been Discarded
A report section header indicating packets that would have been discarded.

messages suppressed
The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

Attack statistics (-A -S) report

This report is displayed when both the -A and -S options are specified on the trmdstat command. It displays the contents of attack statistics records, EZZ8653I. An attack statistics log record contains the number of attacks detected in a specific attack type during a statistics interval. This report takes an attack statistics record and formats it. There is no consolidation or sorting of records. For the Flood type, the attacks number represents the total number of SYN flood and Interface flood starts that are detected during the interval. For the XIDFlood type, the attacks number represents the total number of XID flood starts that are detected during the interval. For the TCPStall type, the attacks number represents the number of times a global TCP stall event is detected during the interval. For the TCPQueSz type, the attacks number represents the total number of TCP queue size constraints that are detected during the interval.

```
>trmdstat -AS /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:19:06 2011

Command Entered     : trmdstat -AS /tmp/tstlog.log
Log Time Interval   : Sep 22 15:08:32  - Nov 29 15:24:28
Stack Time Interval : Sep 22 15:08:22  - Nov 29 19:24:23
TRM Records Scanned : 227

                           ATTACK  Statistics

 Attack           Date and Time              Attacks             Action
--------      ----------------------         ----------         -----------
TCPStall      09/22/2011 15:08:22.06                  0         noresetconn
```

```
TCPQueSz      09/22/2011 15:08:22.07                   0        resetconn
TCPStall      09/22/2011 15:18:14.49                   0        resetconn
EELDLCCk      11/12/2011 04:34:02.05                   0        nodiscard
XIDFlood      11/12/2011 04:34:02.05                   1        nodiscard
EEMalfmd      11/12/2011 05:24:52.34                   3        discard
EEPortCk      11/12/2011 05:24:52.34                   1        discard
Redirect      11/12/2011 18:52:16.19                   0        nodiscard
PerpEcho      11/14/2011 16:03:09.07                   1        nodiscard
NextHdrs      11/18/2011 16:04:59.46                   2        discard
NextHdrs      11/18/2011 18:28:20.17                   1        nodiscard
Flood         11/23/2011 14:46:27.18                   7        discard
OutRaw6       11/29/2011 19:24:23.33                   1        discard
```

The following information describes the areas of the attack statistics report.

Attack Indicates the attack type. The values that can be displayed are:

- DataHide - Data hiding
- DestOpts - Restricted IPv6 destination option
- EELDLCCk - Enterprise Extender LDLC check
- EEMalfmd - EE malformed packet
- EEPortCk - EE source port check
- Flood - SYN flood and interface flood
- Fragment - IP Fragment
- HopOpts - Restricted IPv6 hop-by-hop option
- IPOption - Restricted IPv4 option
- IPProto - Restricted IPv4 protocol
- Malform - Malformed packet
- NextHdrs - Restricted IPv6 next header
- OutRaw4 - Outbound IPv4 Raw
- OutRaw6 - Outbound IPv6 Raw
- PerpEcho - Perpetual echo
- Redirect- ICMP redirect
- TCPQueSz - TCP queue size
- TCPStall - Global TCP stall
- XIDFlood - EE XID flood

Date and Time

Indicates the date and time at which the statistics information was gathered by the TCP/IP stack.

Attacks

Indicates the number of attacks recorded.

Action

Indicates the action that is configured for the attack type. The possible action values are:

discard

Indicates that packets associated with an attack are discarded.

nodiscard

Indicates that packets associated with an attack are not discarded.

resetconn

Indicates that connections associated with an attack are reset.

noresetconn
Indicates that connections associated with an attack are not reset.

Flood summary (-F) report

This report is displayed when the -F option is specified with the trmdstat command. It displays the summary of all flood events. The information presented in this report is derived from EZZ8650I, EZZ8651I, EZZ8654I, EZZ8655I, EZZ8677I, and EZZ8678I types of syslog messages. Summary data related to SYN floods, interface floods, and EE XID floods is shown in separate sections of the report. Summary data for SYN floods is sorted by IP address and then port. Summary data for interface floods is sorted by interface name. Summary data for EE XID floods is sorted by local IP address

```
>trmdstat -F /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:20:43 2011

Command Entered     : trmdstat -F /tmp/tstlog.log
Log Time Interval   : Nov 11 20:35:01  - Nov 23 14:50:52
Stack Time Interval : Nov 11 20:34:41  - Nov 23 14:50:32
TRM Records Scanned : 227

                      SYN FLOOD  Summary

                                              Local SYN Flood  SYN Flood  SYN Flood
              Local IP Address                Port    Start      End      Duration
--------------------------------------------- ----- ---------- ---------- ----------
0.0.0.0                                         360          1          1        183
192.168.105.25                                  444          1          1        168
192.168.105.25                                  452          1          1        169
::                                              345          1          1         99
2001:db8:0:3:9:42:103:132                       345          1          1        168

                      Interface Flood Summary

                  IFC Flood   IFC Flood   IFC Flood
 Interface Name     Start        End       Duration
----------------  ----------  ----------  ----------
LOSAQDIO4                  1           1         324
OSAQDIO46                  2           2         550

                      EE XID Flood Summary

                                               XID Flood   XID Flood   XID Flood
               Local IP Address                  Start        End       Duration
---------------------------------------------  ----------  ----------  ----------
192.168.105.53                                          1           1         233
2001:db8::9:42:105:53                                   1           1         295
```

The following information describes the areas of the SYN flood summary report.

Local IP Address
Specifies the bound IP address.

Local Port
Specifies the bound port number.

SYN Flood Start
Specifies the number of SYN flood starts.

SYN Flood End
Specifies the number of SYN flood ends.

SYN Flood Duration
Specifies the accumulated duration, in seconds, of the SYN floods that have ended.

The following describes the areas of the interface flood summary report.

Interface Name
Specifies the name of the interface for which an interface flood was detected.

IFC Flood Start
Specifies the number of interface flood starts detected.

IFC Flood End
Specifies the number of interface flood ends detected.

IFC Flood Duration
Specifies the accumulated duration, in seconds, of the interface floods that have ended. Duration is non-zero only if the interface has experienced at least one flood that has ended

The following describes the areas of the EE XID FLOOD summary report.

Local IP address
Specifies the destination IP address for which an EE XID flood was detected.

XID Flood Start
Specifies the number of EE XID flood starts detected.

XID Flood End
Specifies the number of EE XID flood ends detected.

XID Flood Duration
Specifies the accumulated duration of the EE XID floods that have ended in seconds. Duration is non-zero only if the local IP address has experienced at least one flood that has ended.

messages suppressed
The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

Flood detail (-F -D) report

This report is displayed when both the -F and -D options are specified with the trmdstat command. It displays the contents of flood event records. The information that is presented in this report is derived from EZZ8650I, EZZ8651I, EZZ8654I, EZZ8655I, EZZ8677I, and EZZ8678I types of syslog messages.

Data that is related to SYN floods, interface floods, and EE XID floods is shown in separate sections of the report. Data for SYN floods and EE XID floods is sorted by IP address. Data for interface floods is sorted by interface name. For the interface flood exit and continuing record types, some information about the discarded packets is also provided. This information includes the protocol discarded most frequently during the flood and the category of discards seen most frequently during the interface flood. If the interface type provides the source MAC address of the prior hop, the most frequently seen prior hop source MAC address is also provided.

```
>trmdstat -FD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Dec  2 14:09:41 2011

Command Entered     : trmdstat -FD /tmp/tstlog.log
Log Time Interval   : Nov 11 20:35:01  - Nov 23 14:50:52
Stack Time Interval : Nov 11 20:34:41  - Nov 23 14:50:32
TRM Records Scanned : 227

                                           SYN FLOOD  Events
```

```
               Date and Time/                Local
               Local IP Address               Port  Type SYNsRecvd   FirstAck  SYNsDiscd  SYNsTimeO   Duration  Correlator
--------------------------------------------- ----- ---- ---------- ---------- ---------- ---------- ---------- ----------
11/20/2011 18:18:15.58                          360   E                                                               4536
  0.0.0.0
11/20/2011 18:21:18.96                          360   X         29          0         29          1        183       4536
  0.0.0.0
11/21/2011 14:59:57.18                          452   E                                                               4583
  192.168.105.25
11/21/2011 15:02:46.79                          452   X        197          0        194        257        169       4583
  192.168.105.25
11/21/2011 16:59:39.97                          444   E                                                               4586
  192.168.105.25
11/21/2011 17:02:28.24                          444   X        198          0        195        257        168       4586
  192.168.105.25
11/21/2011 19:26:42.40                          345   E                                                               4610
  ::
11/21/2011 19:28:21.93                          345   X        499          0        495        257         99       4610
  ::
11/21/2011 18:41:44.76                          345   E                                                               4589
  2001:db8:0:3:9:42:103:132
11/21/2011 18:44:33.71                          345   X        198          1        195        256        168       4589
  2001:db8:0:3:9:42:103:132

                                           Interface FLOOD  Events

   Date and Time/          Interface       Type  Duration    Discard   Correlator/ ----------------Most Frequent---------------------
  Last     Last Source IP/                                   Count/    ProbeID     -----Overall-----   -------Source MAC Data-------
  Count    Dest Address                                      Percent               Proto/  Category/      SrcMAC/  Proto/  Category/
                                                                                   Percent Percent        Percent  Percent Percent
11/22/2011 00:53:07.29     LOSAQDIO4         E                  1000       4751
           192.168.105.50                                         89   04070010
           192.168.105.25
11/22/2011 00:58:09.65     LOSAQDIO4         C        266      21022       4751         6     Queue  000D602432AE       6     Queue
     20023 192.168.105.50                                         95   04070011        95        94            95     100        99
           192.168.105.25
11/22/2011 00:59:10.70     LOSAQDIO4         X        324      21022       4751         6     Queue  000D602432AE       6     Queue
     20023 192.168.105.50                                         95   04070014        95        94            95     100        99
           192.168.105.25
11/22/2011 00:53:29.78     OSAQDIO46         E                  1000       4752
           2001:db8::20a:5eff:fe04:8f16                           94   04070010
           2001:db8::4039:900:540e:3d0
11/22/2011 00:58:33.62     OSAQDIO46         C        269      16814       4752         6     Queue  00062A714400       6     Queue
     15815 2001:db8::20a:5eff:fe04:8f16                           92   04070011        94        93            93     100        99
           2001:db8::4039:900:540e:3d0
11/22/2011 00:59:33.69     OSAQDIO46         X        325      16821       4752         6     Queue  00062A714400       6     Queue
     15822 2001:db8::20a:5eff:fe04:8f16                           79   04070014        94        93            93     100        99
           2001:db8::4039:900:540e:3d0
11/23/2011 14:46:31.78     OSAQDIO46         E                  1000       4832
           2001:db8::20a:5eff:fe04:8f16                          100   04070010
           2001:db8::4039:900:610e:3d0
11/23/2011 14:50:32.28     OSAQDIO46         X        225       6018       4832         6     Queue  00062A714400       6     Queue
      5019 2001:db8::20a:5eff:fe04:8f16                           51   04070014        83        73            83     100        88
           2001:db8::4039:900:610e:3d0

                                           XID FLOOD  Events

                                    Local IP Address/                  -----XID timeouts-----     Last
     Date and Time               Last Source IP Address          Type  Threshold      Flood       Count     Duration   Correlator
----------------------  ---------------------------------------  ----  ----------  ----------  ----------  ----------  ----------
11/11/2011 20:34:41.48  192.168.105.53                             E            2                       3                      36
                        192.168.105.50
11/11/2011 20:38:34.53  192.168.105.53                             X                       15          17         233          36
                        192.168.105.50
11/12/2011 03:53:55.49  2001:db8::9:42:105:53                      E            2                      14                      43
                        2001:db8::20a:5eff:fe04:8f16
11/12/2011 03:58:50.37  2001:db8::9:42:105:53                      X                       13          26         295          43
                        2001:db8::20a:5eff:fe04:8f16
```

The following information describes the areas of the SYN flood detail report.

Date and Time
Specifies the date and time.

Local IP Address
Specifies the bound IP address.

Local Port
Specifies the bound port number.

Type Specifies the entry to or exit from constrained state.

E enter

X exit

SYNsRecvd
The number of handshakes started during SYN flood. Present only on EXIT records.

FirstAck
The number of handshakes completed during SYN flood. Present only on EXIT records.

SYNsDiscd
The number of SYNs randomly discarded during SYN flood. Present only on EXIT records.

SYNsTimeO
The number of SYNs timing out during SYN flood. Present only on EXIT records.

Duration
Specifies the duration of flood in seconds. Present only on EXIT records.

Correlator
Specifies the trace correlator.

The following describes the areas of the interface flood events report.

Date and Time
Specifies the date and time.

Interface
The name of the interface experiencing the interface flood condition.

Type Specifies flood entry, flood exit, or continuing flood condition.

E enter

X exit

C continuing

Duration
The number of seconds since the start of the interface flood was detected. Duration is displayed in both continuing and exit records.

Discard Count/Percent

Discard Count
On interface flood entry, this is the number of discarded inbound packets or not processed packets that triggered the interface flood detection. On interface flood exit or continuation, this is the number of inbound packets discarded or not processed since the interface flood was detected.

Discard Percent
On interface flood entry, this is the percentage of total packets received that were discarded and that triggered the interface flood detection. On interface flood exit or continuation, this is the

percentage of total packets received that were discarded on the interface since the interface flood was detected.

Correlator/ProbeID

Correlator
Specifies the trace correlator.

ProbeID
Specifies the IDS probeID that generated this event.

Last Count
The consecutive number of discarded packets for the interface that have the same source IP address as the last discarded packet. If the previously discarded packet's source IP address is not the same as the last discarded packet's source IP address, the count is one. Reported for interface flood continuing and exit record types.

Last Source IP/Dest Address

Last Source IP address
Source IP address of the last packet discarded on this interface during the interface flood condition.

Destination Address
Local IP address associated with the interface when the interface flood was detected.

Most Frequent
This data is tracked from the time the interface flood is detected until the interface flood ends. The counts do not include the initial discards that contributed to the interface flood detection.

This data is reported for interface flood continuing and exit record types. The data is cumulative from the time the interface flood started until the time the record was generated.

Overall

Proto/Percent

Proto IP protocol most frequently seen in the discarded packets. The protocol value is the protocol number or zero if the protocol value is invalid or unknown.

Percent
Percentage of times the protocol was seen in the discarded packets.

Category/Percent

Category
Discard category most frequently seen in the discarded packets. Possible values are:

Storage
Storage could not be obtained to process the packet. Storage shortages might indicate a problem in the system other than an inbound packet flood.

CheckSum
Packet had checksum error.

Malform
Malformed packet.

Dest Destination not found. For example, the port is not active or is reserved, the matching socket not available, no listeners for the RAW protocol.

Firewall
Packet rejected by IP security.

MedHdr
Bad media header.

Forward
Packet is not for us but could not be forwarded. Some cases that prevent forwarding are bad headers or IPCONFIG NODATAGRAMFWD specified.

QOSPol
Packet dropped due to QoS policy.

IDSPol
Packet dropped due to IDS policy.

Access
Packet dropped due to NetAccess, multilevel security, or OSM access checks.

ATTLS
Packet dropped due to AT-TLS policy.

OtherPol
Packet dropped due to other configuration policy.

Queue
Queue limit (other than those specified by IDS) prevented queueing the packet for processing. For example, the SYN queue, the reassembly queue, the UDP or RAW receive queues.

OtherSyn
Syn problems other than SYN queue full.

State State mismatch.

UnpackEr
Packet dropped due to unpacking problems.

Misc Miscellaneous reasons not listed above. For example, TCP packet outside of TCP window, duplicate fragments found during packet reassembly.

Percent
The percentage of times the discard category was seen in the discarded packets.

Source MAC Data
Source MAC Data is reported for LCS devices and OSA QDIO

devices at a microcode level that supports providing the source MAC address of the prior hop. It is not applicable for other devices. This data is reported for interface flood continuing and exit record types.

SrcMAC/Percent

SrcMAC
Source MAC of the prior hop seen most frequently in the discarded packets. The value `N/A` appears in the field if the device does not support providing the source MAC.

Percent
Percentage of times the most frequent source MAC was seen in the discarded packets.

Proto/Percent

Proto The most frequent IP protocol seen in the discarded packets associated with the source MAC address. The protocol value is the protocol number or zero if the protocol value is invalid or unknown.

Percent
Percentage of times the protocol was seen in the discarded packets associated with the source MAC address.

Category/Percent

Category
The most frequent discard category seen in the discarded packets associated with the source MAC address. The possible values are the same as those listed for Most Frequent Overall Category.

Percent
Percentage of times the discard category was seen in the discarded packets associated with the source MAC address.

The following list describes the areas of the EE XID FLOOD detail report.

Date and Time
Specifies the date and time.

Local IP Address
Specifies the destination IP address of the XID flood.

Last Source IP Address
Source IP address of the last XID that timed out to this local IP address during the EE XID flood condition.

Type Specifies the entry to or exit from constrained state.

E Enter. Use type E for the XID timeout threshold, which is the number of XID timeouts that occurred before an EE XID flood was detected.

X Exit. Use type X for the number of XIDs that timed out during this EE XID flood.

XID Timeouts

Last Count

The consecutive number of XID timeouts to the local IP address that has the same source IP address as the last XID that timed out. If the previously timed out XID source IP address is not the same as the last XID time-out packet's source IP address, the count is 1.

Duration

Specifies the duration of the flood in seconds. Duration is displayed only on exit records.

Correlator

Specifies the trace correlator.

messages suppressed

The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

The interface flood events report width is 132 characters. If you are displaying or printing this report, use an output device that can accommodate this width.

Flood statistics (-F -S) report

This report is displayed when both the -F and -S options are specified on the trmdstat command. It displays the contents of attack flood statistics records only. This report only formats an attack flood statistics record. There is no consolidation or sorting of records. An overall flood statistics log record, EZZ8653I with attack type Flood, contains the number of floods detected during a statistics interval regardless of the type of flood.

More detailed statistics information is also kept by interface for interface flood reporting and to provide data to help an installation determine the policy action values for flood percentage and minimum discard that are used for interface flood detection. The interface flood specific statistics information is contained in the EZZ8657I statistics record and is reported in the Interface FLOOD Detailed Statistics section of the report.

More detailed statistics information is also kept by local IP address for EE XID flood reporting. The EE XID flood specific statistics information is contained in the EZZ8676I statistics record and is reported in the XID FLOOD Detailed Statistics section of the report.

```
>trmdstat -FS /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:24:56 2011

Command Entered     : trmdstat -FS /tmp/tstlog.log
Log Time Interval   : Nov 11 20:37:31  - Nov 23 15:48:22
Stack Time Interval : Nov 11 20:37:15  - Nov 23 15:48:07
TRM Records Scanned : 227

                  Overall FLOOD Statistics

    Date and Time        Flood Count
----------------------   ----------
11/23/2011 14:46:27.18            7

                  Interface FLOOD Detailed Statistics

    Date and Time         Interface    -----Discard-----       Attacks
                                           Count       Pct
---------------------- ----------------  ----------    ---    ----------
11/23/2011 14:46:27.18 LOSAQDIO4              14943     24             1
11/23/2011 15:48:07.51 LOSAQDIO4              82122     74             1
```

```
11/23/2011 14:46:27.18 OSAQDIO46               1756     18             0
11/23/2011 15:48:07.51 OSAQDIO46               8231     26             1

                  XID FLOOD Detailed Statistics

                                                                       -----XID Timeouts-----
     Date and Time                     Local IP Address                 Interval      Peak       Attacks
----------------------  ---------------------------------------------  ----------  ----------  ----------
11/11/2011 20:37:15.56  192.168.104.196                                        20           2           0
11/11/2011 20:37:15.56  192.168.105.53                                         40           4           3
11/11/2011 20:37:15.56  2001:db8::9:42:105:53                                  10           1           0
11/12/2011 03:54:27.58  2001:db8::9:42:105:53                                   4           1           1
11/12/2011 04:04:37.57  2001:db8::9:42:105:53                                  12           4           0
```

The following information describes the areas of the overall flood statistics report.

Date and Time
Indicates the date and time at which the statistics information was gathered by the TCP/IP stack.

Flood Count
The total number of SYN flood and Interface flood entries detected during the interval.

The following describes the areas of the interface flood detailed statistics report.

Date and Time
Indicates the date and time at which the statistics information was gathered by the TCP/IP stack.

Interface
The name of the interface for which the data is reported.

Discard Count
Number of inbound packets discarded or not processed during the statistics interval.

Discard Pct
Percentage of the total packets received on the interface during the statistics interval that were discarded.

Attacks
Number of Interface flood entries detected on the interface during the statistics interval.

The following list describes the areas of the EE XID FLOOD detailed statistics report.

Date and Time
Indicates the date and time at which the statistics information was gathered by the TCP/IP stack.

Local IP Addres
Destination IP address for which the data is reported.

Timeout Interval
Number of inbound EE XID packets that timed out during the statistics interval.

Timeout Peak
The maximum number of EE XID packets that timed out during a 1-minute interval.

Attacks
: Number of EE XID floods starts detected during the statistics interval.

Global TCP stall summary (-G) report

This report is displayed when the -G option is specified on the trmdstat command. It displays the summary of global TCP stall events. The information presented in this report is derived from EZZ8671I, EZZ8672I, EZZ8673I, and EZZ8674I types of syslog messages. In the Connections Reset and Connections Would Have Been Reset sections, information is sorted by remote IP address.

```
>trmdstat -G /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 13:16:37 2011

Command Entered     : trmdstat -G /tmp/tstlog.log
Log Time Interval   : Oct 29 18:02:33  - Oct 29 18:53:33
Stack Time Interval : Oct 29 18:02:19  - Oct 29 18:53:21
TRM Records Scanned : 504

               Global TCP Stall Summary

Global TCP Stall Entered:  1
Global TCP Stall Exited:   1
Global TCP Stall Duration: 2920

               Connections Reset

No records to display

               Connections Would Have Been Reset

            Remote IP Address             Count
--------------------------------------- ----------
10.11.2.1                                      126
10.12.2.1                                      125
2001:db8:10::11:2:1                            126
2001:db8:10::12:2:1                            125
```

The following information describes the areas of the global TCP stall summary report:

Global TCP Stall Entered
: The number of global TCP stall enter conditions that have been detected.

Global TCP Stall Exited
: The number of global TCP stall exit conditions that have been detected.

Global TCP Stall Duration
: The accumulated duration, in seconds, of the global TCP stall conditions that have ended. Global TCP Stall Duration is non-zero if the TCP/IP stack has experienced at least one global TCP stall that has ended.

Remote IP Address
: The remote IP address of one or more stalled TCP connections.

Count The number of stalled TCP connections with the remote IP address that contributed to a global TCP stall condition. When the global TCP stall condition was detected, the connections were reset if a reset action was requested in the Intrusion Detection Services (IDS) policy for the global TCP stall attack type.

messages suppressed
: The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in "The trmdstat report general concept" on page 844 for a detailed description.

Global TCP stall detail (-G -D) report

This report is displayed when both the -G and -D options are specified on the trmdstat command. It displays the contents of individual global TCP stall event records. The information presented in this report is derived from EZZ8671I, EZZ8672I, EZZ8673I, and EZZ8674I types of syslog messages. In the Connections Reset and Connections Would Have Been Reset sections, information is grouped and sorted by remote IP address.

```
>trmdstat -GD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Tue Dec  6 13:16:51 2011

Command Entered     : trmdstat -GD /tmp/tstlog.log
Log Time Interval   : Oct 29 18:02:33  - Oct 29 18:53:33
Stack Time Interval : Oct 29 18:02:19  - Oct 29 18:53:21
TRM Records Scanned : 504

                             Global TCP Stall Events

                                                   Small    Write
                                Stalled            Window   Block
    Date and Time        Type   Percent TotalConns Percent Percent  Duration  Correlator   Action
---------------------- -------- ------- ---------- ------- ------- ---------- ---------- -----------
10/29/2011 18:02:19.08 Enter        50%       1004     49%      0%                     3 noresetconn
10/29/2011 18:53:21.22 Exit         25%       2008     25%      0%       2920          3 noresetconn

                             Connections Reset

No records to display

                             Connections Would Have Been Reset

Remote IP Address: 10.11.2.1

                       Local                                         Remote ConnID/  SendQSize/
    Date and Time      Port               Local IP Address           Port   JobName  WindowSize Correlator
---------------------- ----- --------------------------------------- -----  -------- ---------- ----------
10/29/2011 18:02:19.08 20000 10.11.1.2                                1119  00000091       8000          3
                                                                            USER13            0
10/29/2011 18:02:19.08 20000 10.11.1.2                                1140  000000A5      20000          3
                                                                            USER13            0

Remote IP Address: 10.12.2.1

                       Local                                         Remote ConnID/  SendQSize/
    Date and Time      Port               Local IP Address           Port   JobName  WindowSize Correlator
---------------------- ----- --------------------------------------- -----  -------- ---------- ----------
10/29/2011 18:02:19.08 20000 10.12.1.2                                1117  0000008F      16000          3
                                                                            USER13            0
10/29/2011 18:02:19.08 20000 10.12.1.2                                1513  0000021C       9500          3
                                                                            USER13            0

Remote IP Address: 2001:db8:10::11:2:1

                       Local                                         Remote ConnID/  SendQSize/
    Date and Time      Port               Local IP Address           Port   JobName  WindowSize Correlator
---------------------- ----- --------------------------------------- -----  -------- ---------- ----------
10/29/2011 18:02:19.09 25000 2001:db8:10::11:1:2                      1456  000001EA      10000          3
                                                                            USER22            0
10/29/2011 18:02:19.09 25000 2001:db8:10::11:1:2                      1352  00000182       8000          3
                                                                            USER22            0


Remote IP Address: 2001:db8:10::12:2:1

                       Local                                         Remote ConnID/  SendQSize/
    Date and Time      Port               Local IP Address           Port   JobName  WindowSize Correlator
```

```
---------------------- ----- --------------------------------------- -----  -------- ---------- ----------
10/29/2011 18:02:19.09 25000 2001:db8:10::12:1:2                      1310  00000158      20000          3
                                                                            USER22            0
10/29/2011 18:02:19.09 25000 2001:db8:10::12:1:2                      1102  00000088      16000          3
                                                                            USER22            0
```

The following information describes the areas of the global TCP stall event detail report:

Date and Time
The stack date and time when the event occurred.

Type

Enter A global TCP stall condition was entered

Exit A global TCP stall condition was exited.

ExitPlcy
A global TCP stall condition was exited because IDS Global TCP Stall policy was no longer in effect.

Stalled Percent
The percentage of the active TCP connections that were stalled. A TCP connection is considered stalled if one or more of the following conditions are true:

- The TCP send window size is less than 256 or is less than the smaller of the largest send window that has been seen for the connection and the default MTU. The TCP send window size is set based on values provided by the TCP peer. The default MTU for IPv4 is 576. The default MTU for IPv6 is 1280.
- The TCP send queue is full and the data is not being retransmitted.

TotalConns
The total number of active TCP connections.

Small Window Percent
The percentage of the active TCP connections that were stalled because the TCP send window size is less than the smaller of the MSS of the connection and the default MTU. A TCP connection can be stalled due to multiple conditions. For example, a TCP connection might be included in both the Small Window Percent value and the Write Block Percent value.

Write Block Percent
The percentage of the active TCP connections that were stalled because the TCP send queue is full and the data is not being retransmitted. A TCP connection can be stalled due to multiple conditions. For example, a TCP connection might be included in both the Small Window Percent value and the Write Block Percent value.

Duration
The duration, in seconds, of the global TCP stall. Present only for Exit and ExitPlcy records.

Correlator
The correlator for a global TCP stall condition. The correlator can be used to correlate global TCP stall enter and exit conditions with individual TCP connections that contributed to the condition. The individual connections are reset or would have been reset.

Action

The action specified in the IDS policy for the global TCP stall attack type. The action value can be resetconn or noresetconn.

Results:

- If the value is resetconn, all stalled TCP connections were reset. If you requested detailed syslogd messages for the global TCP stall attack type in the IDS policy, the Connections Reset section contains an entry for each stalled connection that was reset during the global TCP stall attack.
- If the value is noresetconn, stalled TCP connections were not reset. However, if you requested detailed syslogd messages for the global TCP stall attack type in the IDS policy, the Connection Would Have Been Reset section contains an entry for each connection that was stalled at the time that the global TCP stall was detected.

Remote IP Address

The remote IP address of the stalled TCP connections described in the table.

Local Port

The local port number of the stalled TCP connection that was reset or would have been reset.

Local IP Address

The local IP address of the stalled TCP connection that was reset or would have been reset.

Remote Port

The remote port number of the stalled TCP connection that was reset or would have been reset.

ConnID

The ID of the TCP connection that was reset or would have been reset.

JobName

The job name of the TCP connection that was reset or would have been reset.

SendQSize

The amount of data queued to send queue for the stalled TCP connection.

WindowSize

The size of the send window for the stalled TCP connection.

messages suppressed

The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

IDS overall summary (-I) report option

This report is displayed when the -I option is specified with the trmdstat command or when no report option is provided on the trmdstat command invocation. It displays the summary of all the IDS information present in the log. Using this report enables you to get an idea of the overall effect of the IDS policies installed in the system.

```
> trmdstat  -I /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 11:30:11 2011

Command Entered     : trmdstat -I /tmp/tstlog.log
Log Time Interval   : Jul 19 10:41:39  - Nov 23 14:52:52
Stack Time Interval : Jul 19 10:41:39  - Nov 23 14:52:31
```

```
TRM Records Scanned : 227

TCP - Traffic Regulation
------------------------------------------------
Connections would have been refused :          3
Connections refused                 :         19

Constrained entry logged            :          1
Constrained exit logged             :          1
Constrained entry                   :          5
Constrained exit                    :          5

QOS exceptions logged               :          1
QOS exceptions made                 :          5

UDP - Traffic Regulation
------------------------------------------------
Constrained entry logged            :          1
Constrained exit logged             :          1
Constrained entry                   :          4
Constrained exit                    :          4

SCAN Detection
------------------------------------------------
Threshold exceeded                  :         11
Detection delayed                   :          0
Storage constrained entry           :          0
Storage constrained exit            :          0

ATTACK Detection
------------------------------------------------
Packet would have been discarded    :         10
Packet discarded                    :         12

FLOOD Detection
------------------------------------------------
Accept queue expanded               :          2
SYN flood start                     :          5
SYN flood end                       :          5
Interface flood start               :          3
Interface flood end                 :          3
EE XID flood start                  :          2
EE XID flood end                    :          2

Global TCP Stall Detection
------------------------------------------------
Global TCP stall entry              :          1
Global TCP stall exit               :          1
Connections would have been reset   :          6
Connections reset                   :          6

TCP Queue Size Detection
------------------------------------------------
Send queue
  Constrained entry                 :          2
  Constrained exit                  :          2
  Connections reset                 :          2
Receive queue
  Constrained entry                 :          2
  Constrained exit                  :          2
  Connections reset                 :          2
Out-of-order queue
  Constrained entry                 :          2
  Constrained exit                  :          2
  Connections reset                 :          1
```

The following information describes the areas of the IDS summary report.

TCP - Traffic regulation

Connections would have been refused
Specifies the number of connections that would have been refused if policy action LIMIT had been specified in the TR policy. This count indicates the total number of EZZ9319I messages present in the log.

Connections refused
Indicates the number of connections refused by the system. This count indicates the total number of EZZ9324I messages present in the log.

Constrained entry logged
Specifies the number of times that a TCP listener would have entered a constrained state if policy action LIMIT had been specified in the TR policy. This count indicates the total number of EZZ9320I messages present in the log.

Constrained exit logged
Specifies the number of times that a TCP listener would have exited a constrained state if policy action LIMIT had been specified in the TR policy. This count indicates the total number of EZZ9322I messages present in the log.

Constrained entry
Specifies the number of times that a TCP listener entered a constrained state. This count indicates the total number of EZZ9321I messages present in the log.

Constrained exit
Specifies the number of times that a TCP listener exited a constrained state. This count indicates the total number of EZZ9323I messages present in the log.

QOS exceptions logged
Specifies the number of times a QoS exception was logged because the QOS policy guarantees a higher number of connections to this port than would be allowed by the TCP TR policy. This count indicates the total number of EZZ9318I messages present in the log.

QOS exceptions made
Specifies the number of times a QoS exception was made because the QOS policy guarantees a higher number of connections to this port than would be allowed by the TCP TR policy. This count indicates the total number of EZZ9317II messages present in the log.

UDP - Traffic regulation

Constrained entry logged
Specifies the number of times that a UDP socket would have entered a constrained state if policy action LIMIT had been specified in the TR policy. This count indicates the total number of EZZ8638I messages present in the log.

Constrained exit logged
Specifies the number of times that a UDP socket would have exited

a constrained state if policy action LIMIT had been specified in the TR policy. This count indicates the total number of EZZ8640I messages present in the log.

Constrained entry
Specifies the number of times that a UDP socket entered a constrained state. This count indicates the total number of EZZ8639I messages present in the log.

Constrained exit
Specifies the number of times that a UDP socket exited a constrained state. This count indicates the total number of EZZ8641I messages present in the log.

Scan detection

Threshold exceeded
Specifies the number of scan events detected. This count indicates the total number of EZZ8643I messages present in the log.

Detection delayed
Specifies the number of scan interval overrun events detected. This count indicates the total number of EZZ8645I messages present in the log.

Storage constrained entry
Specifies the number of times scan storage constraint entry was detected. This count indicates the total number of EZZ8646I messages present in the log.

Storage constrained exit
Specifies the number of times scan storage constraint exit was detected. This count indicates the total number of EZZ8647I messages present in the log.

Attack detection

Packet would have been discarded
Specifies the total number of attack packets that would have been discarded if policy action Discard had been specified in the attack policy. This count indicates the total number of EZZ8649I messages present in the log.

Packet discarded
Specifies the total number of attack packets discarded. This count indicates the total number of EZZ8648I messages present in the log.

Flood detection

Accept queue expanded
Specifies the number of accept queue expansions. This count indicates the total number of EZZ8652I messages present in the log.

SYN flood start
Specifies the number of SYN flood starts detected. This count indicates the total number of EZZ8650I messages present in the log.

SYN flood end

Specifies the number of SYN flood ends detected. This count indicates the total number of EZZ8651I messages present in the log.

Interface flood start

Specifies the number of interface flood starts detected. This count indicates the total number of EZZ8654I messages present in the log.

Interface flood end

Specifies the number of interface flood ends detected. This count indicates the total number of EZZ8655I messages present in the log.

EE XID flood start

Specifies the number of EE XID flood starts detected. This count indicates the total number of EZZ8677I messages present in the log.

EE XID flood end

Specifies the number of EE XID flood ends detected. This count indicates the total number of EZZ8678I messages present in the log.

Global TCP stall detection

Global TCP stall entered

Specifies the number of global TCP stall enter conditions that have been detected. This count indicates the number of EZZ8671I messages present in the log.

Global TCP stall exited

Specifies the number of global TCP stall exit conditions that have been detected. This count indicates the number of EZZ8672I messages present in the log.

Connections would have been reset

Specifies the number of stalled TCP connections that contributed to a global TCP stall condition. These stalled TCP connections were not reset because Intrusion Detection Services (IDS) policy for the global TCP stall attack type specified that connections should not be reset. This count indicates the number of EZZ8674I messages present in the log.

Connections reset

Specifies the number of stalled TCP connections that contributed to a global TCP stall condition. These stalled TCP connections were reset because Intrusion Detection Services (IDS) policy for the global TCP stall attack type specified that connections should be reset. This count indicates the number of EZZ8673I messages present in the log.

TCP Queue size detection

Send queue

Constrained entered

Specifies the number of times a TCP connection's send queue entered a constrained state. This count indicates the total number of EZZ8664I messages present in the log.

Constrained exited

Specifies the number of times a TCP connection's send queue exited a constrained state. This count indicates the total number of EZZ86651I messages present in the log.

Connections reset

Specifies the number of TCP connections that were reset because the connections' send queues were constrained. This count indicates the number of EZZ8669I messages present in the log.

Receive queue

Constrained entered

Specifies the number of times a TCP connection's receive queue entered a constrained state. This count indicates the total number of EZZ8662I messages present in the log.

Constrained exited

Specifies the number of times a TCP connection's receive queue exited a constrained state. This count indicates the total number of EZZ86631I messages present in the log.

Connections reset

Specifies the number of TCP connections that were reset because the connections' receive queues were constrained. This count indicates the number of EZZ8668I messages present in the log.

Out-of-order queue

Constrained entered

Specifies the number of times a TCP connection's out-of-order queue entered a constrained state. This count indicates the total number of EZZ8666I messages present in the log.

Constrained exited

Specifies the number of times a TCP connection's out-of-order exited a constrained state. This count indicates the total number of EZZ86671I messages present in the log.

Connections reset

Specifies the number of TCP connections that were reset because the connections' out-of-order queues were constrained. This count indicates the number of EZZ8670I messages present in the log.

messages suppressed

Specifies the number of messages suppressed with date and time. This data comes from an EZZ8660I, EZZ8661I, or EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

Scan summary (-N) report

This report is displayed when the -N option is specified on the trmdstat command. It displays the summary of scan events. The information presented in this report is derived from EZZ8643I type syslog messages. The information is sorted by source IP address.

```
> trmdstat -N /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 08:34:49 2011
```

```
Command Entered     : trmdstat -N /tmp/tstlog.log
Log Time Interval   : Jul 19 10:41:39  - Jul 23 12:54:15
Stack Time Interval : Jul 19 10:41:39  - Jul 23 16:54:06
TRM Records Scanned : 128

                                         SCAN  Summary

              Source IP Address                       Scans                 Suspicion Level
                                                 Fast       Slow       Very     Possibly    Normal
--------------------------------------------- ---------- ---------- ---------- ---------- ----------
192.168.16.48                                          1          1          8         10         22
2001:db8:0:a:209:6bff:fee9:65dd                        3          2          6         11          6
2001:db8:11:16::44                                     1          1         10         35         19
2001:db8:11:16:202:55ff:fe31:148c                      1          1         15          0         11
```

The following information describes the areas of the scan summary report.

Source IP Address
Specifies the IP address of the source host that triggered scan detection.

Fast Scans
Specifies the number of fast scans detected.

Slow Scans
Specifies the number of slow scans detected.

Suspicion Level
Specifies the number of packets at each suspicion level that contributed to the scan detection.

Restriction: When a scan is detected for a source IP address, additional suspicious packets from that source IP that are received during the current fast scan interval are not reflected in these suspicious counts.

Very Specifies the number of packets at the very suspicious suspicion level that contributed to the scan detection.

Possible
Specifies the number of packets at the possibly suspicious suspicion level that contributed to the scan detection.

Normal
Specifies the number of packets at the normal suspicion level that contributed to the scan detection.

Scan detail (-N -D) report

This report is displayed when both the -N and -D options are specified on the trmdstat command. It displays the contents of individual scan event records. The records are sorted by source IP address. The information in this report is derived from EZZ8643I type syslog messages.

```
> trmdstat -ND /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 08:35:40 2011

Command Entered     : trmdstat -ND /tmp/tstlog.log
Log Time Interval   : Jul 19 10:41:39  - Jul 23 12:54:15
Stack Time Interval : Jul 19 10:41:39  - Jul 23 16:54:06
TRM Records Scanned : 128

                                         SCAN  Events

    Date and Time                    Source IP Address                       Suspicion Level         Type Correlator
                                                                        Very     Possibly    Normal
---------------------- --------------------------------------------- ---------- ---------- ---------- ---- ----------
07/22/2011 15:23:22.34 192.168.16.48                                          8          0         12   S          35
07/22/2011 16:12:27.55 192.168.16.48                                          0         10         10   F          55
07/19/2011 10:41:39.63 2001:db8:0:a:209:6bff:fee9:65dd                        0          1          2   F           2
07/19/2011 15:14:40.96 2001:db8:0:a:209:6bff:fee9:65dd                        0          3          0   F          20
```

```
07/19/2011 15:36:40.09 2001:db8:0:a:209:6bff:fee9:65dd                        3          3          1   S          23
07/19/2011 20:41:39.07 2001:db8:0:a:209:6bff:fee9:65dd                        0          1          2   F          32
07/19/2011 25:36:40.09 2001:db8:0:a:209:6bff:fee9:65dd                        3          3          1   S          33
07/23/2011 13:16:34.04 2001:db8:11:16::44                                     0         19          0   F          62
07/23/2011 16:54:06.04 2001:db8:11:16::44                                    10         16         19   S          65
07/22/2011 15:30:05.34 2001:db8:11:16:202:55ff:fe31:148c                      6          0          0   F          38
07/22/2011 16:02:07.53 2001:db8:11:16:202:55ff:fe31:148c                      9          0         11   S          42
```

The following information describes the areas of the scan detail report.

Date and Time
Specifies the date and time in the message at which the scan events were logged.

Source IP Address
Specifies the IP address of the source host that triggered scan detection.

Suspicion Level
Specifies the number of packets at each suspicion level that contributed to the scan detection.

Restriction: When a scan is detected for a source IP address, additional suspicious packets from that source IP that are received during the current fast scan interval are not reflected in these suspicious counts.

Very Specifies the number of packets at the very suspicious suspicion level that contributed to the scan detection.

Possible
Specifies the number of packets at the possibly suspicious suspicion level that contributed to the scan detection.

Normal
Specifies the number of packets at the normal suspicion level that contributed to the scan detection.

Type Specifies the scan type.

F Fast

S Slow

Correlator
Specifies the trace correlator.

TCP queue size (-Q) summary report

This report is displayed when the -Q option is specified on the trmdstat command. It displays the summary of TCP queue size events. The information presented in this report is derived from EZZ8662I, EZZ8663I, EZZ8664I, EZZ8665I, EZZ8666I, EZZ8667I, EZZ8668I, EZZ8669I, and EZZ8670I types of syslog messages. Information is sorted by remote IP addresses, and then by local ports.

```
>trmdstat -Q /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:30:03 2011

Command Entered     : trmdstat -Q /tmp/tstlog.log
Log Time Interval   : Nov 10 15:24:11  - Nov 11 13:42:34
Stack Time Interval : Nov 10 15:23:50  - Nov 11 13:42:10
TRM Records Scanned : 227

                 TCP Queue Size Summary

                   Connections Reset

Remote IP Address: 2001:db8::20a:5eff:fe04:8f16

Local
Port  SendQReset RecvQReset OofOQReset
----- ---------- ---------- ----------
```

```
 1000          0          0          1

Remote IP Address: 2001:db8:10::11:1:2

Local
Port  SendQReset RecvQReset OofOQReset
----- ---------- ---------- ----------
 1001          1          0          0

                   TCP Queue Constraints

Remote IP Address: 2001:db8::20d:60ff:fe24:32ae

Local ------- SendQ Constraint ------- ------- RecvQ Constraint ------- ------- OofOQ Constraint -------
Port    Enter       Exit     Duration    Enter       Exit     Duration    Enter       Exit     Duration
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
 1000          0          0          0          1          1       3541          1          1       1044

Remote IP Address: 2001:db8:10::11:1:2

Local ------- SendQ Constraint ------- ------- RecvQ Constraint ------- ------- OofOQ Constraint -------
Port    Enter       Exit     Duration    Enter       Exit     Duration    Enter       Exit     Duration
----- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
 1000          1          1        756          0          0          0          0          0          0
 1001          1          1        497          0          0          0          0          0          0
```

The following information describes the areas of the TCP queue size summary report:

Remote IP Address
The remote IP address of the TCP connection. This value will be an IP address on the local TCP/IP stack when the IP addresses at both ends of the TCP connection reside on the local TCP/IP stack.

Local Port
The local port number of the TCP connection.

SendQReset
The number of TCP connections that were reset because there was either excessive or old data accumulating on the connection's send queue.

RecvQReset
The number of TCP connections that were reset because there was either excessive or old data accumulating on the connection's receive queue.

OofOQReset
The number of TCP connections that were reset because there was either excessive or old data accumulating on the connection's out-of-order queue.

SendQ Constraint

Enter The number of times that a TCP connection's send queue entered constrained state because there was either excessive or old data accumulating on the queue.

Exit The number of times that a TCP connection's send queue exited constrained state.

Duration
Specifies the accumulated duration, in seconds, that a TCP connection's send queue was constrained. Duration is non-zero only if at least one constraint condition has exited.

RecvQ Constraint

Enter The number of times that a TCP connection's receive queue entered constrained state because there was either excessive or old data accumulating on the queue.

Exit The number of times that a TCP connection's receive queue exited constrained state.

Duration
Specifies the accumulated duration, in seconds, that a TCP connection's receive queue was constrained. Duration is non-zero only if at least one constraint condition has exited.

OofOQ Constraint

Enter The number of times that a TCP connection's out-of-order queue entered constrained state because there was either excessive or old data accumulating on the queue.

Exit The number of times that a TCP connection's out-of-order queue exited constrained state.

Duration
Specifies the accumulated duration, in seconds, that a TCP connection's out-of-order queue was constrained. Duration is non-zero only if at least one constraint condition has exited.

messages suppressed
The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in "The trmdstat report general concept" on page 844 for a detailed description.

TCP queue size (-Q -D) detail report

This report is displayed when both the -Q and -D options are specified on the trmdstat command. It displays the contents of individual TCP Queue Size event records. The information presented in this report is derived from EZZ8662I, EZZ8663I, EZZ8664I, EZZ8665I, EZZ8666I, EZZ8667I, EZZ8668I, EZZ8669I, and EZZ8670I types of syslog messages. Information is grouped and sorted by remote IP addresses.

```
>trmdstat -QD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 25 09:31:27 2011

Command Entered     : trmdstat -QD /tmp/tstlog.log
Log Time Interval   : Nov 10 15:24:11  - Nov 11 13:42:34
Stack Time Interval : Nov 10 15:23:50  - Nov 11 13:42:10
TRM Records Scanned : 227

                                            TCP Queue Size Events

                                              Connections Reset

Remote IP Address: 2001:db8::20a:5eff:fe04:8f16

                             Local                                         Remote ConnID/  QueueSize/  DageAge/
    Date and Time      Queue Port               Local IP Address           Port   JobName   Trigger    BytesQed  Correlator
---------------------- ----- ----- --------------------------------------- ------ -------- ---------- ---------- ----------
11/10/2011 24:39:34.18 OofO   1000 2001:db8::9:42:105:17                    54224 000000D1     S             30          26
                                                                                  IBMUSER1  BytesQed      25600

Remote IP Address: 2001:db8:10::11:1:2

                             Local                                         Remote ConnID/  QueueSize/  DageAge/
    Date and Time      Queue Port               Local IP Address           Port   JobName   Trigger    BytesQed  Correlator
---------------------- ----- ----- --------------------------------------- ------ -------- ---------- ---------- ----------
11/10/2011 15:58:50.33 Send   1001 2001:db8:10::11:2:1                       1001 0000009C     S            110          25
                                                                                  IBMUSER1  BytesQed       1000


                                              TCP Queue Constraints

Remote IP Address: 2001:db8::20d:60ff:fe24:32ae

                       Type/ Local                                         Remote ConnID/  QueueSize/  DageAge/   Duration/
    Date and Time      Queue Port               Local IP Address           Port   JobName   Trigger    BytesQed  Correlator
```

```
---------------------- ----- ----- --------------------------------------- ------ -------- ---------- ---------- ----------
11/10/2011 17:33:27.02 Enter  1000 2001:db8::9:42:105:17                    61572 000000A5     S            132
                       OofO                                                       IBMUSER3  DataAge        6840          5
11/10/2011 17:51:41.87 Exit   1000 2001:db8::9:42:105:17                    61572 000000A5     S              0       1044
                       OofO                                                       IBMUSER3                    0          5
11/10/2011 18:38:27.35 Enter  1000 2001:db8::9:42:105:17                    60468 0000012E     S             97
                       Recv                                                       IBMUSER8  DataAge        4080         12
11/10/2011 19:40:20.24 Exit   1000 2001:db8::9:42:105:17                    60468 0000012E     S              0       3541
                       Recv                                                       IBMUSER8                    0         12

Remote IP Address: 2001:db8:10::11:1:2

                       Type/ Local                                         Remote ConnID/  QueueSize/  DageAge/   Duration/
    Date and Time      Queue Port               Local IP Address           Port   JobName   Trigger    BytesQed  Correlator
---------------------- ----- ----- --------------------------------------- ------ -------- ---------- ---------- ----------
11/10/2011 23:40:37.49 Enter  1000 2001:db8:10::11:2:1                       1001 0000003D     S             41
                       Send                                                       IBMUSER1  BytesQed       1025         21
11/10/2011 23:53:49.90 Exit   1000 2001:db8:10::11:2:1                       1001 0000003D     S              0        756
                       Send                                                       IBMUSER1                    0         21
11/10/2011 23:58:43.53 Enter  1001 2001:db8:10::11:2:1                       1001 0000003D     S            113
                       Send                                                       IBMUSER1  DataAge         515         22
11/10/2011 24:07:24.82 Exit   1001 2001:db8:10::11:2:1                       1001 0000003D     S              0        497
                       Send                                                       IBMUSER1                    0         22
```

The following information describes the areas of the TCP Queue Size detail report:

Remote IP Address
The remote IP address of the TCP connection associated with the event. This value will be an IP address on the local TCP/IP stack when the IP addresses at both ends of the TCP connection reside on the local TCP/IP stack.

Date and Time
The stack date and time when the event occurred.

Type Event type. Possible values include:

Enter Constrained state was entered because there was either excessive or old data accumulating on the queue.

Exit Constrained state was exited.

Queue
The TCP connection's queue associated with the event. Possible values include:

OofO Out-of-order queue

Recv Receive queue

Send Send queue

Local Port
The local port number of the TCP connection associated with the event.

Local IP Address
The local IP address of the TCP connection associated with the event.

Remote Port
The remote port of the TCP connection associated with the event.

ConnID
The connection ID of the TCP connection associated with the event.

JobName
The job name of the TCP connection associated with the event.

QueueSize
The abstract TCP queue size value that was configured in the IDS policy for the TCP queue size attack type.

VS Very short

S Short

L Long

VL Very long

Trigger
The condition that triggered the event. This field is present only for Enter type records. The possible values are:

DataAge
The event was triggered by the length of time that data had been on the specified queue.

BytesQed
The event was triggered by the amount of data on the specified queue.

DataAge
The age in seconds of the oldest data on the specified queue when the event occurred.

BytesQd
The number of bytes queued on the specified queue when the event occurred.

Duration
The number of seconds that the specified queue was constrained. This field is present only for Exit type records.

Correlator
The correlator for a constrained queue condition. The correlator can be used to correlate TCP queue size constrained enter and exit events.

messages suppressed
The number of attack messages suppressed with attack type, date and time. This data comes from an EZZ9327I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

TCP TR summary (-T) report

This report is displayed when the -T option is specified with the trmdstat command. It displays the summary of all TCP traffic regulation events. The information presented in this report is derived from EZZ9317I, EZZ9318I, EZZ9319I, EZZ9320I, EZZ9321I, EZZ9322I, EZZ9323I, and EZZ9324I types of syslog messages. Information is grouped by local host and sorted by local host and then by local port.

```
>trmdstat -T /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:01:09 2011

Command Entered     : trmdstat -T /tmp/tstlog.log
Log Time Interval   : May 27 19:19:23  - May 28 21:26:23
Stack Time Interval : May 27 19:19:05  - May 28 21:26:02
TRM Records Scanned : 86

                          TCP TR Summary

Local Host: ::

              Constrained States                      Connections
Local              Limited                Excp          Refused
Port    Enter       Exit     Duration     QOS        Appl       Host
----- ---------- ---------- ---------- ---------- ---------- ----------
```

```
  21         0         0          0         0         0         1
 333         1         1        277         0         0         6
 345         2         2       2308         3         1         9

No TypeActions LOG records to display
```

The following information describes the areas of the TR TCP Summary Report:

Local Host
: If the policy was configured to limit by all sockets (also known as limit by port), this is either 255.255.255.255 or ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff. If the policy was configured to limit by each socket (also known as limit by port instance), this is the IP address that is bound to by the local listener applications. The value 0.0.0.0 indicates that the application bound to INADDR_ANY. The value :: indicates that the application bound to in6addr_any.

Limited | Logged
: For each Local Host the report is first generated for ports with a policy that specified both an action of LIMIT and an action of LOG and then for ports that specified only a policy action of LOG.

Local Port
: Indicates the port number bound to by a local listener application.

Constrained States
: The number of times this port entered and exited constrained state and the total duration in seconds of constrained state.

Excp QOS
: The number of connections that were allowed because the QoS policy for a particular source IP guaranteed a higher number of connections to this port than were allowed by the TCP TR percentage of total connections for a single host.

Connections Refused Appl
: The number of connections refused because the total number of connections limit was exceeded.

Connections Refused Host
: The number of connections refused because the number of connections requested from a single host exceeded the percentage of total connections allowed for a single host.

Connections Would Have Been Refused Appl
: The number of connections that would have been refused because the total number of connections limit was exceeded. The connections were allowed because policy action LIMIT was not specified.

Connections Would Have Been Refused Host
: The number of connections that would have been refused because the number of connections requested from a single host exceeded the percentage of total connections allowed for a single host. The connections were allowed because policy action LIMIT was not specified.

messages suppressed
: The number of TCP TR messages suppressed with date and time, type of suppressed TCP connections, local IP address, local listening port, and the reason that the log records were suppressed. This data comes from an EZZ8660I or EZZ8661I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

TCP TR detail (-T -D) report

This report is displayed when both the -T and -D options are specified with the trmdstat command. It displays the contents of individual TCP TR records. The information displayed in this report is derived from EZZ9317I, EZZ9318I, EZZ9319I, EZZ9320I, EZZ9321I, EZZ9322I, EZZ9323I, and EZZ9324I types of syslog messages. Information is grouped and sorted by local host.

```
>trmdstat -TD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:01:38 2011

Command Entered     : trmdstat -TD /tmp/tstlog.log
Log Time Interval   : May 28 20:40:53  - May 28 21:26:23
Stack Time Interval : May 28 20:40:48  - May 28 21:26:02
TRM Records Scanned : 86

                                         TCP TR Events

                                         Events Limited

Local Host: ::

    Date and Time/     Local Rec Cns      Connections               Policy
      Source Host      Port  Typ Typ  Current   Available  Total Conn Pct Qos Limit  Correlator ProbeID
---------------------- ----- --- --- ---------- ---------- ---------- --- ---------- ---------- --------
05/28/2011 20:40:48.88   345  Q               3          7         10  50         10          1  1004014
  10.42.105.25
05/28/2011 20:41:49.14   345  Q               4          6         10  50         10          1  1004014
  10.42.105.25
05/28/2011 20:59:08.62   345  S   E           0          1         10  50          0          3  1004400
  2001:db8::9:42:105:25
05/28/2011 21:01:26.98   345  C               4          1         10  50          0          2  1004044
  2001:db8::9:42:105:25
05/28/2011 21:05:23.97   345  C               1          0         10  50          0          3  1004048
  10.42.105.135
05/28/2011 21:26:02.29   345  S   X           0          2         10  50          0          3  1002400
  2001:db8::9:42:105:25

No TypeActions LOG records to display
```

The following information describes the areas of the TCP TR detail report:

Events Limited | Logged
For each Local Host the report is first generated for ports with a policy that specifies both an action of LIMIT and an action of LOG and then for ports that specify only an action of LOG.

Local Host
If the policy was configured to limit by all sockets (also known as limit by port), this is either 255.255.255.255 or ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff. If the policy was configured to limit by each socket (also known as limit by port instance), this is the IP address bound to by the local listener applications. The value 0.0.0.0 indicates that the application bound to INADDR_ANY. The value :: indicates that the application bound to in6addr_any.

Date and Time
The stack date and time the event occurred.

Local Port
The port bound to by a local listener application.

Source Host
The source host associated with the event.

Rec Typ
The record type of the event. Possible values are:

- C - Connection refused or would have been refused events

- Q - Connection allowed due to QoS exception events. The QOS policy guarantees a higher number of connections to this port than would be allowed by the TCP TR policy
- S - Port entered or exited constraint events

Cns Typ
Constraint event type:
- E - Entered
- X - Exited

Connections Current
The number of connections that were active on this port at the time of this event and that were established while policy was in effect.

Connections Available
The remaining number of connections available to this port at the time of this event.

Policy Total Conn
The total number of connections allowed for this port.

Policy Pct
The percentage of total connections that can be used by a single IP address for this port.

Policy QoS Limit
The maximum number of connections specified in the QoS policy for this source host and this port.

Correlator
The trace correlator for this event.

ProbeID
The IDS probeID that generated this event.

messages suppressed
The number of TCP TR messages suppressed with date and time, type of suppressed TCP connections, local IP address, local listening port, and the reason that the log records were suppressed. This data comes from an EZZ8660I or EZZ8661I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

TCP TR extended (-T -E) report

This report is displayed when both the -T and -E options are specified with the trmdstat command. It displays an extended summary of all TCP traffic regulation events. For each port a separate line of totals is generated for each source host. The information presented in this report is derived from EZZ9317I, EZZ9318I, EZZ9319I, EZZ9320I, EZZ9321I, EZZ9322I, EZZ9323I, and EZZ9324I types of syslog messages. Information is sorted by local host and then by local port and source host pair.

```
>trmdstat -TE /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:03:02 2011

Command Entered     : trmdstat -TE /tmp/tstlog.log
Log Time Interval   : May 27 19:19:23  - May 28 21:26:23
Stack Time Interval : May 27 19:19:05  - May 28 21:26:02
TRM Records Scanned : 86

                                         TCP TR Extended Summary

Local Host: ::
                                                           Constrained States                       Connections
Local                                                           Limited                 Excp           Refused
```

```
Port                Source Host                       Enter       Exit     Duration     QOS        Appl       Host
----- -------------------------------------------- ---------- ---------- ---------- ---------- ---------- ----------
   21 10.65.201.199                                         0          0          0          0          0          1
  333 10.42.0.1                                             1          1        277          0          0          1
  333 10.42.105.25                                          0          0          0          0          0          1
  333 10.42.105.135                                         0          0          0          0          0          1
  333 2001:db8::9:42:105:25                                 0          0          0          0          0          1
  333 2001:db8:0:1:9:42:105:135                             0          0          0          0          0          2
  345 10.42.0.1                                             1          0          0          0          0          0
  345 10.42.105.25                                          0          0          0          2          0          2
  345 10.42.105.135                                         0          0          0          0          1          2
  345 2001:db8::9:42:105:25                                 1          1       1613          1          0          2
  345 2001:db8:0:1:9:42:105:135                             0          1        695          0          0          3

No TypeActions LOG records to display
```

The following information describes the areas of the TR TCP extended summary report:

Local Host
If the policy was configured to limit by all sockets (also known as limit by port), this is either 255.255.255.255 or ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff. If the policy was configured to limit by each socket (also known as limit by port instance), this is the IP address that is bound to by the local listener applications. The value 0.0.0.0 indicates that the application bound to INADDR_ANY. The value :: indicates that the application bound to in6addr_any.

Limited | Logged
For each Local Host the report is first generated for ports with a policy that specified both an action of LIMIT and an action of LOG and then for ports that specified only a policy action of LOG.

Local Port
Indicates the port number bound to by a local listener application.

Source Host
For each port a separate line of totals is generated for each source host.

Constrained States
The number of times this port entered and exited constrained state and the total duration in seconds of constrained state.

Excp QOS
The number of connections that were allowed because the QoS policy for a particular source IP guaranteed a higher number of connections to this port than were allowed by the TCP TR percentage of total connections for a single host.

Connections Refused Appl
The number of connections refused because the total number of connections limit was exceeded.

Connections Refused Host
The number of connections refused because the number of connections requested from a single host exceeded the percentage of total connections allowed for a single host.

Connections Would Have Been Refused Appl
The number of connections that would have been refused because the total number of connections limit was exceeded. The connections were allowed because policy action LIMIT was not specified.

Connections Would Have Been Refused Host
The number of connections that would have been refused because the

number of connections requested from a single host exceeded the percentage of total connections allowed for a single host. The connections were allowed because policy action LIMIT was not specified.

messages suppressed
: The number of TCP TR messages suppressed with date and time, type of suppressed TCP connections, local IP address, local listening port, and the reason that the log records were suppressed. This data comes from an EZZ8660I or EZZ8661I message. See in “The trmdstat report general concept” on page 844 for a detailed description.

TCP TR statistics (-T -S) report

This report is displayed when both the -T and -S options are specified on the trmdstat command. It displays the contents of the TCP traffic regulation statistics records, EZZ9316I. Information is sorted by local host.

```
>trmdstat -TS /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:02:30 2011

Command Entered     : trmdstat -TS /tmp/tstlog.log
Log Time Interval   : May 28 20:43:53  - May 28 21:25:23
Stack Time Interval : May 28 20:43:40  - May 28 21:25:00
TRM Records Scanned : 86

                                              TCP TR Statistics

Local Host: ::

    Date and Time/         Local                   Peak/    Requests/   Warnings/   QosExcepts/ Terminates/
      Peak Host            Port      Action      HostPeak    Current     Duration    SugLimit   SugPercent
----------------------     -----     -------    ----------  ----------  ----------  ----------  ----------
05/28/2011 20:43:40.45       345     LIMIT               5           5           0           2           0
  10.42.105.25                                           5           5           0           0           0
05/28/2011 20:59:10.46       345     LIMIT               9           4           0           2           0
  2001:db8::9:42:105:25                                  4           9           1           0           0
05/28/2011 21:04:20.49       345     LIMIT              10           2           1           0           1
  2001:db8::9:42:105:25                                  4          10         310           0           0
05/28/2011 21:09:30.51       345     LIMIT              10           1           1           0           1
  10.42.105.135                                          1          10         310           0           0
05/28/2011 21:14:40.59       345     LIMIT              10           0           0           0           0
  0.0.0.0                                                0          10         310           0           0
05/28/2011 21:19:50.58       345     LIMIT              10           0           0           0           0
  0.0.0.0                                                0          10         309           0           0
05/28/2011 21:25:00.57       345     LIMIT              10           0           0           0           1
  10.42.105.25                                           4           9         309           0           0
```

The following information describes the areas of the TR TCP statistics report:

Local Host
: If the policy was configured to limit by all sockets (also known as limit by port), this is either 255.255.255.255 or ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff. If the policy was configured to limit by each socket (also known as limit by port instance), this is the IP address bound to by the local listener applications. The value 0.0.0.0 indicates that the application bound to INADDR_ANY. The value :: indicates that the application bound to in6addr_any.

Date and Time
: The stack date and time the statistics were reported.

Local Port
: The port bound to by a local listener application.

Action
: Indicates whether or not an action of LIMIT was specified in the policy that is in effect at the end of the statistics interval.

Peak The highest number of concurrent connections from all sources during the statistics interval.

Peak Host
The IP address of the source host with the largest number of concurrent connections that also requested an additional connection during the statistics interval.

HostPeak
The number of allowed connections held by the source host identified in Peak Host.

Requests
The total number of new connection requests received during the statistics interval.

Warnings
The total number of connections that would have been denied during the statistics interval if a policy action of LIMIT had been in effect at the time of the request.

QosExcepts
The total number of connections that were allowed during this statistics interval because the QoS policy for a particular source IP guaranteed a higher number of connections to this port than were allowed by the TCP TR percentage of total connections for a single host.

Terminates
The total number of connections that were denied during the statistics interval because a policy action of LIMIT was in effect at the time of the request.

Current
The number of connections existing at the end of the statistics interval.

Duration
The number of seconds this port was in constrained state during this statistics interval.

SugLimit
A suggested value for the total number of connections limit that will avoid any connections being denied for exceeding either limit in future periods with the same number of total requests and requests from a single source. If a policy action of LIMIT was in effect then this value is 0.

SugPercent
A suggested companion value for the percentage of total connections that can be used by a single IP address. If a policy action of LIMIT was in effect then this value is 0.

TCP TR connection (-C) report

This report is displayed when the -C option is specified with the trmdstat command. It displays the connection summary information of all TCP traffic regulation events. The information presented in this report is derived from EZZ9319I and EZZ9324I types of syslog messages. Information is grouped and sorted by local port..

```
>trmdstat -C /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:03:57 2011

Command Entered     : trmdstat -C /tmp/tstlog.log
Log Time Interval   : May 27 19:19:23  - May 28 21:05:53
Stack Time Interval : May 27 19:19:05  - May 28 21:05:23
TRM Records Scanned : 86

Local Port  Connections Refused
```

```
----------  -------------------
    21                 1
   333                 6
   345                10

Local Port  Connections Would Have Been Refused
----------  -----------------------------------
No records to display
```

The following information describes the areas of the TCP TR connection summary report:

Local Port
Indicates the port number bound to by a local listener application.

Connections Refused
The total number of connections refused for this port.

Connections Would Have Been Refused
The total number of connections that would have been refused for this port if a policy action of LIMIT was specified in the policy.

TCP TR connection detail (-C -D) report

This report is displayed when both the -C and -D options are specified with the trmdstat command. It displays the connection information of all TCP traffic regulation events. The information presented in this report is derived from EZZ9319I and EZZ9324I types of syslog messages. Information is grouped and sorted by local port and then by source IP address.

```
>trmdstat -CD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:04:15 2011

Command Entered     : trmdstat -CD /tmp/tstlog.log
Log Time Interval   : May 27 19:19:23  - May 28 21:05:53
Stack Time Interval : May 27 19:19:05  - May 28 21:05:23
TRM Records Scanned : 86

Local Port  Connections Refused              Source IP Address
----------  -------------------  ---------------------------------------------
    21                 1
                            1   10.65.201.199
   333                 6
                            1   10.42.0.1
                            1   10.42.105.25
                            1   10.42.105.135
                            1   2001:db8::9:42:105:25
                            2   2001:db8:0:1:9:42:105:135
   345                10
                            2   10.42.105.25
                            3   10.42.105.135
                            2   2001:db8::9:42:105:25
                            3   2001:db8:0:1:9:42:105:135

             Connections Would
Local Port   Have Been Refused               Source IP Address
----------  -------------------  ---------------------------------------------
No records to display
```

This report first shows the total number of 'Connections Refused' or 'Connection Would Have Been Refused' by port number. Under each port total, the data for the port is broken out by source IP for which the connection was refused or would have been refused.

Data under the 'Connections Refused' heading reports on connections that were refused because the policy specified an action of limit. Data under the 'Connections Would Have Been Refused' heading are connections that exceeded policy limits but were not rejected because policy did not specify an action of limit.

The following information describes the areas of the TR TCP connection detail report:

Local Port
Indicates the port number bound to by a local listener application.

Connections Refused
If the report line contains a Port Number, this is the total number of connections refused for this port. If the report line contains an IP Address, this is the number of connections refused for the Port and IP Address combination.

Connections Would Have Been Refused
If the report line contains a Port Number, then this is the total number of connections that would have been refused for this port if an action of LIMIT was specified in the policy. If the report line contains an IP Address, this is the number of connections that would have been refused for this port and IP Address combination if an action of LIMIT was specified in the policy.

Source IP Address
Indicates the source IP address.

TCP TR connection detail for a specific host (-C -D -h) report

This report is displayed when -h filter is specified along with both the -C and -D options in the trmdstat command. It displays the connection information for the particular host whose IP address matches to the -h filter value. The information presented in this report is derived from EZZ9319I and EZZ9324I types of syslog messages. Information is grouped and sorted by local port.

```
>trmdstat -CD -h 10.42.105.25 /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Wed Dec 14 09:05:09 2011

Command Entered     : trmdstat -CD -h 10.42.105.25 /tmp/tstlog.log
Log Time Interval   : May 27 21:15:49  - May 28 01:39:08
Stack Time Interval : May 27 21:15:38  - May 28 01:39:03
TRM Records Scanned : 86

Local Port  Connections Refused
----------  -------------------
                Date and Time       HostCurrent   Available   Percent
            ----------------------  -----------   ----------  -------
    333     05/27/2011 21:15:38.67            2            3    50%
    345     05/28/2011 01:38:39.75            2            3    50%
    345     05/28/2011 01:39:03.85            2            3    50%

Local Port  Connections Would Have Been Refused
----------  -----------------------------------
                Date and Time       HostCurrent   Available   Percent
            ----------------------  -----------   ----------  -------
No records to display
```

The following information describes the areas of the TCP TR connection detail when the -h filter is used to retrieve data for a specific host:

Local Port
Indicates the port number bound to by a local listener application.

Connections Refused
Each entry under this heading represents a connection request that was refused.

Connections Would Have Been Refused
Each entry under this heading represents a connection that would have been refused if a policy action of LIMIT was specified.

Date and Time
The stack date and time the connection was refused or would have been refused.

HostCurrent
The number of connections that were active on this port at the time the connection was refused or would have been refused.

Available
The remaining number of connections available to this port at the time the connection was refused or would have been refused.

Percent
The percentage of total connections that can be used by a single IP address for this port.

UDP TR summary (-U) report

This report is displayed when the -U option is specified with the trmdstat command. It displays the summary of UDP constrained state and datagram discard information. The information presented in this report is derived from EZZ8638I, EZZ8639I, EZZ8640I, and EZZ8641I syslog messages. Information is grouped and sorted by local IP address and then by local port.

```
> trmdstat -U /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 11 08:54:00 2011

Command Entered     : trmdstat -U /tmp/tstlog.log
Log Time Interval   : Nov  9 22:15:10  - Nov 10 03:02:22
Stack Time Interval : Nov  9 22:14:59  - Nov 10 03:02:19
TRM Records Scanned : 128

                                         UDP TR Summary

                                               Local        Constrained State         Datagrams
              Local IP Address                 Port   Entered     Exited   Duration   Discarded
---------------------------------------------- ----- ---------- ---------- ---------- ----------
192.168.105.41                                   601          1          1       1273       2809
2001:db8::1                                      700          1          1       2164          5
2001:db8::1                                      701          1          1        246          5
2001:db8::105:41                                 801          1          1         88        845

                                                                                      Datagrams
                                                                                      Would
                                               Local        Constrained State         Have Been
              Local IP Address                 Port   Entered     Exited   Duration   Discarded
---------------------------------------------- ----- ---------- ---------- ---------- ----------
2001:db8:0:1:9:42:105:174                        901          1          1        260          0


TRMD Started for TCPCS   : Oct 28 01:23:18
```

The following information describes the areas of the UDP summary report.

Local IP Address
Specifies the bound IP address.

Local Port
Specifies the bound port number.

Constrained State

Specifies constrained state status.

Entered

The number of constrained state entries.

Exited The number of constrained state exits.

Duration

Specifies the accumulated duration, in seconds, that the UDP inbound queue was constrained based on the duration reported when a constraint was exited. Duration is non-zero only if the UDP inbound queue exited a constraint at least once.

Datagram Disposition

Specifies disposition of datagrams.

Discarded

Specifies the number of datagrams discarded.

Would Have Been Discarded

Specifies the number of datagrams that would have been discarded if policy action LIMIT had been specified in UDP TR policy.

UDP TR detail (-U -D) report

This report is displayed when both the -U and -D options are specified. It displays the contents of individual UDP records. The information presented in this report is derived from EZZ8638I, EZZ8639I, EZZ8640I, and EZZ8641I syslog messages. Information is grouped and sorted by local IP address.

```
> trmdstat -UD /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 11 08:54:56 2011

Command Entered     : trmdstat -UD /tmp/tstlog.log
Log Time Interval   : Nov  9 22:15:10  - Nov 10 03:02:22
Stack Time Interval : Nov  9 22:14:59  - Nov 10 03:02:19
TRM Records Scanned : 128

                             UDP TR Events

Local IP Address: 192.168.105.41

                       Local
    Date and Time      Port  Type  Duration  Discarded  Qsize Correlator
---------------------- ----- ---- ---------- ---------- ----- ----------
11/09/2011 23:32:45.49   601  E                          VL            6
11/09/2011 23:53:59.28   601  X         1273       2809  VL            6

Local IP Address: 2001:db8::1

                       Local
    Date and Time      Port  Type  Duration  Discarded  Qsize Correlator
---------------------- ----- ---- ---------- ---------- ----- ----------
11/09/2011 22:14:59.33   700  E                          VS            3
11/09/2011 22:39:38.61   701  E                          VS            4
11/09/2011 22:43:44.66   701  X          246          5  VS            4
11/09/2011 22:51:03.96   700  X         2164          5  VS            3

Local IP Address: 2001:db8::105:41

                       Local
    Date and Time      Port  Type  Duration  Discarded  Qsize Correlator
---------------------- ----- ---- ---------- ---------- ----- ----------
11/09/2011 23:20:21.41   801  E                           S            5
11/09/2011 23:21:49.52   801  X           88        845   S            5
```

```
Local IP Address: 2001:db8:0:1:9:42:105:174

                                               Would
                       Local                 Have Been
    Date and Time      Port  Type  Duration  Discarded  Qsize Correlator
---------------------- ----- ---- ---------- ---------- ----- ----------
11/10/2011 02:57:59.01   901  E                           L           25
11/10/2011 03:02:19.78   901  X          260          0   L           25


TRMD Started for TCPCS   : Oct 28 01:23:18
```

The following information describes the areas of the UDP detail report.

Local IP Address
Specifies the bound IP address.

Date and Time
Specifies the date and time the event occurred.

Local Port
Specifies the bound port number.

Type Specifies the entry to or exit from constrained state. Add the values in a list under Type.

E Enter

X Exit

Duration
Specifies the duration of constrained state in seconds. This field is present only on EXIT records.

Datagram Disposition
Specifies the datagram disposition. This field is present only on EXIT records.

- Number of datagrams Discarded
- Number of datagrams that Would have been Discarded if policy action LIMIT had been specified in UDP TR policy.

Qsize Specifies the queue limit specified on the policy.

VS Very small

S Small

L Large

VL Very large

Correlator
Specifies the trace correlator.

UDP TR statistics (-U -S) report

This report is displayed when both the -U and -S options are specified on the trmdstat command. It displays the contents of the UDP traffic regulation statistics records, EZZ8642I. There is no consolidation or sorting of records.

```
> trmdstat -US /tmp/tstlog.log
trmdstat for z/OS CS V2R1  Fri Nov 11 08:56:06 2011

Command Entered     : trmdstat -US /tmp/tstlog.log
Log Time Interval   : Nov  9 23:07:10  - Nov 10 03:50:53
Stack Time Interval : Nov  9 23:06:56  - Nov 10 03:50:48
TRM Records Scanned : 128
```

```
                                       UDP TR Statistics

Local IP Address: 192.168.105.41

                             Datagrams Received/    Bytes Received/
                       Local Datagrams Discarded/   Bytes Discarded/
    Date and Time      Port  Datagrams Peak         Bytes Peak          Duration  Constraints Qsize Action
---------------------- ----- -------------------- -------------------- ---------- ----------- ----- -------
11/10/2011 00:28:37.31   601                 8192              8192000       1273           1  VL   LIMIT
                                             2809              2809000
                                             8191              8191000
11/10/2011 00:28:37.31   600                    0                    0          0           0  VL   LIMIT
                                                0                    0
                                                0                    0

Local IP Address: 2001:db8::1

                             Datagrams Received/    Bytes Received/
                       Local Datagrams Discarded/   Bytes Discarded/
    Date and Time      Port  Datagrams Peak         Bytes Peak          Duration  Constraints Qsize Action
---------------------- ----- -------------------- -------------------- ---------- ----------- ----- -------
11/09/2011 23:06:56.90   701                   16                15100        246           1  VS   LIMIT
                                                5                 5000
                                               15                15000
11/09/2011 23:06:56.90   700                   15                15000       2164           1  VS   LIMIT
                                                5                 5000
                                               15                15000

Local IP Address: 2001:db8::105:41

                             Datagrams Received/    Bytes Received/
                       Local Datagrams Discarded/   Bytes Discarded/
    Date and Time      Port  Datagrams Peak         Bytes Peak          Duration  Constraints Qsize Action
---------------------- ----- -------------------- -------------------- ---------- ----------- ----- -------
11/10/2011 00:18:37.22   800                    0                    0          0           0   S   LIMIT
                                                0                    0
                                                0                    0
11/10/2011 00:18:37.22   801                  256               256000         88           1   S   LIMIT
                                              845               845000
                                              255               255000

Local IP Address: 2001:db8:0:1:9:42:105:174

                             Datagrams Received/    Bytes Received/
                       Local Datagrams Discarded/   Bytes Discarded/
    Date and Time      Port  Datagrams Peak         Bytes Peak          Duration  Constraints Qsize Action
---------------------- ----- -------------------- -------------------- ---------- ----------- ----- -------
11/10/2011 03:50:48.62   900                 1000              1000000          0           0   L   NOLIMIT
                                                0                    0
                                              540               540000
11/10/2011 03:50:48.62   901                 2001              2001000        260           1   L   NOLIMIT
                                                0                    0
                                             2000              2000000

TRMD Started for TCPCS   : Oct 28 01:23:18
```

The following describes the areas of the UDP statistics report.

Local IP Address
Specifies the bound IP address.

Date and Time
Specifies the date and time in the message when the statistics were logged.

Local Port
Specifies the bound port number.

Datagrams Received
Specifies the number of datagrams received in the statistics interval.

Datagrams Discarded
Specifies the number of datagrams that were discarded or would have been discarded during the statistics interval. If Action is LIMIT, then this is

the number of datagrams discarded. If Action is NOLIMIT, then this is the number of datagrams that would have been discarded.

Datagrams Peak
Specifies the largest number of datagrams queued during the statistics interval. Set only if a receive is processed during the statistics interval. Does not include datagrams from a Pascal API.

Bytes Received
Specifies the number of bytes received in the statistics interval.

Bytes Discarded
Specifies the number of bytes that were discarded or would have been discarded during the statistics interval. If Action is LIMIT, then this is the number of bytes discarded. If Action is NOLIMIT, then this is the number of bytes that would have been discarded.

Bytes Peak
Specifies the largest number of bytes queued during the statistics interval. Set only if a receive is processed during the statistics interval.

Duration
Specifies the number of seconds the UDP inbound queue was constrained during this statistics interval.

Constraints
Specifies the number of times the UDP inbound queue entered the constrained state during this statistics interval.

Qsize Specifies the queue size specified on the policy.

VS Very small

S Small

L Large

VL Very large

Action
LIMIT if the policy action LIMIT has been specified in UDP TR policy. NOLIMIT if the policy action LIMIT has not been specified in the UDP TR policy.

Chapter 6. Querying and administrating a Domain Name System (DNS)

This information describes the Domain Name System (DNS) domain names, domain name servers, resolvers, and resource records. It also provides the following descriptions:

- NSLOOKUP, **onslookup**, **nsupdate**, DIG, and **dig** commands used to query name servers
- **hostname**, **dnsdomainname**, and **domainname** commands used to display the local DNS host name and domain name

Resolver related commands

Programs that query a name server are called resolvers. Because many TCP/IP applications need to query the name server, a set of routines is usually provided for application programmers to perform queries. However, utility programs with resolver interface are provided for system administrators to interactively query and update the name server.

z/OS Communications Server provides the following resolver related utility programs:

- NSLOOKUP, see "Using the TSO NSLOOKUP command"
- **onslookup**/**nslookup**, see "Using the z/OS UNIX onslookup/nslookup command" on page 905
- nsupdate, see "Using the z/OS UNIX nsupdate command" on page 916
- DIG (TSO), see "Using the TSO DIG command" on page 921
- **dig** (z/OS UNIX), see "Using the z/OS UNIX dig command" on page 937
- host, see "Using the z/OS UNIX host command" on page 947

The TSO DIG command uses the z/OS Communications Server provided resolver for resolver facilities. The BIND 9 **onslookup** and **dig** commands use the resolver initialization facilities of the z/OS Communications Server provided resolver but use their own resolver for additional resolver facilities needed. For a complete discussion of resolver configuration files, see the z/OS Communications Server: IP Configuration Guide.

Restriction: Scope information is not permitted on the operands that represent the target host name on the NSLOOKUP, **onslookup**, **nslookup**, **nsupdate**, DIG (TSO), or **dig** (z/OS UNIX) utility programs.

Using the TSO NSLOOKUP command

The NSLOOKUP command enables you to query name servers in order to accomplish the following tasks:

- Locate information about network nodes
- Examine the contents of a name-server database
- Establish the accessibility of name servers

Rule: The NSLOOKUP command does not use resolver caching.

NSLOOKUP has two modes of operation: interactive mode and command mode. Interactive mode enables you to repeatedly query one or more name servers for information about various hosts and domains and display that information on your terminal. Command mode displays the output from the query supplied as part of the command and then exits.

TSO NSLOOKUP has been deprecated in favor of the z/OS UNIX **dig** command. There are a number of the more recent resource record types that TSO NSLOOKUP will not understand, including the forward and some reverse resource records used for IPv6.

NSLOOKUP configuration

The configuration options of NSLOOKUP determine the operation and results of your name server queries. You can configure NSLOOKUP operation using the following methods:

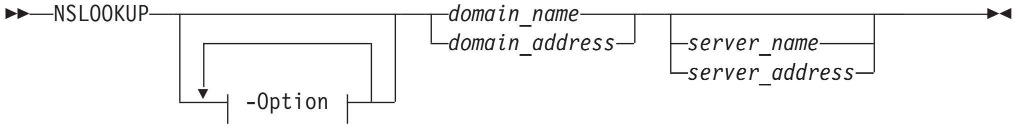
- TCP/IP client program configuration data set, TCPIP.DATA
- NSLOOKUP options data set, *user_id*.NSLOOKUP.ENV
- NSLOOKUP command options

For information about the TCPIP.DATA data set, see the z/OS Communications Server: IP Configuration Reference. For information about the NSLOOKUP.ENV options data set and the NSLOOKUP command options, see "NSLOOKUP options" on page 897.

NSLOOKUP: Query a name server in command mode

Use the NSLOOKUP command to specify an individual query in command mode.

Format



Parameters

-Option
For a description of the NSLOOKUP options, see "NSLOOKUP options" on page 897.

domain_name
Queries the name server for information about the current query type of *domain_name*. The default query type is A (address query).

If the domain name starts with an underscore (_), you must prefix the domain name with the escape character (\).

domain_address
Reverses the components of the address and generates a pointer type (PTR) query to the name server for the `in-addr.arpa` domain mapping of the address to a domain name.

server_name
Directs the default name server to map *server_name* to an IP address and then use the name server at that IP address.

server_address
Specifies the IP address of the name server to be queried other than the default name server. A query for the address in the `in-addr.arpa` domain is initially made to the default name server to map the IP address to a domain name for the server.

Usage

The parameters and subcommands of NSLOOKUP are case sensitive and must be entered in lowercase. Parameter values and domain names are not case sensitive.

If the resolver trace is active, the trace will show the initial values before the NSLOOKUP command line options are processed.

Context

- See "NSLOOKUP options" on page 897 for the complete list and description of NSLOOKUP options.
- See "NSLOOKUP: Issue queries to name servers in interactive mode" for the complete list and description of subcommand and query formats.

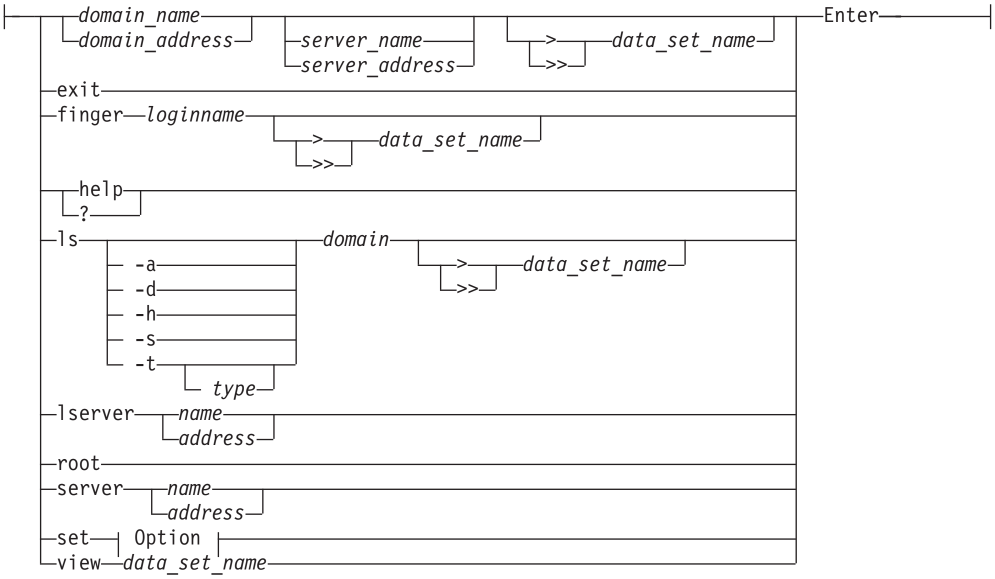
NSLOOKUP: Issue queries to name servers in interactive mode

Use the NSLOOKUP command to issue multiple queries in interactive mode. In interactive mode, an initial query is made to the selected name server to verify that the server is accessible. All subsequent interactive queries are sent to that server unless you specify another server using the *server* or *lserver* options.

Format

```
►►──NSLOOKUP──┬──────────────────┬──┬──────────────────┬──Enter──────────►
              │  ┌────────────┐  │  ├─ - server_name───┤
              │  ▼            │  │  └─ - server_address┘
              └──┤ -Option ├──┴──┘


   ┌────────────────┐
   ▼                │
►──┤ SubCommand ├───┴──────────────────────────────────────────────────►◄
```

SubCommand:



Parameters

Queries processed by NSLOOKUP that specify an address can give unexpected results. If the current query type is address (A) or domain-name pointer (PTR), NSLOOKUP generates a PTR type query for the specified address in the `in-addr.arpa` domain. This returns PTR records which define the host name for the specified address. If the current query type is neither of these two types, a query is performed using the current query type, with the domain name specified as the address given.

Text that does not conform to the defined options and follows the preceding syntax is treated as a domain query. NSLOOKUP does not issue a query for a domain name if the name is unqualified and is the same as one of the defined options.

-Option
For a description of the NSLOOKUP options, see "NSLOOKUP options" on page 897.

address
Specifies the IP address of the server.

data_set_name

Output can be placed in a data set for later viewing by specifying *data_set_name*. The > *data_set_name* option places the output in *data_set_name* and overwrites the contents, if any, of the data set. The >> *data_set_name* option places the output in *data_set_name* and appends it to the contents, if any, of the data set. There must be at least one space before and after the > or >> symbol.

domain_address
Reverses the components of the address and generates a pointer type (PTR) query to the name server for the `in-addr.arpa` domain mapping of the address to a domain name.

domain_name
Queries the name server for information about the current query type of *domain_name*. The default query type is A (address query).

If the domain name starts with an underscore (_), you must prefix the domain name with the escape character (\).

exit
Exits from NSLOOKUP interactive mode.

finger *parms*
Extracts information from the finger server of the node found in the last address query. By default, this command returns a list of logged-in users for the node last found. You can find information about a particular user by specifying the *loginname* of the user as a parameter.

An error occurs if the preceding subcommand was not a successful address query or finger operation. If the current host is not defined, querying the name server defines that name server to be the current host for a subsequent finger operation.

The finger option expects that the finger server is operating on the node found. An error occurs if the server is not operating or the node cannot be reached.

help or ?
Displays a brief summary of commands.

loginname
The logged-in user name. The *loginname* variable is case sensitive and must be specified in the same case (upper or lower) as that used by the host.

ls *parms*
Lists various information available for the domain. By default, the IP address of each node in the domain is listed.

To select resource records other than the default, specify one of the following options:

-a CNAME

-d ALL

-h HINFO

-s WKS

-t [*type*]
Retrieves the resource record type specified in *type*. If no record type is specified with the **-t** option, the current default type is used.

If *type* is ns, up to 24 characters of the returned DNS name is displayed. The UNIX **onslookup**/**nslookup** command can be used to display the entire DNS name.

See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query types.

The **ls** command expects the domain name specified in *domain* to be a zone. If the domain name specified refers to a host, an error message is printed and no information is given. This command should create a virtual circuit (TCP connection) with the current name server to service the request. An error message is printed if the virtual circuit cannot be established.

A # symbol is displayed at the terminal as every 50 lines are written to the data set to indicate the command is still executing.

lserver *parms*
Changes the current server. If *server_name* is specified, the IP address of *server_name* is determined using the initial server defined at command invocation.

An error occurs if the domain name cannot be mapped to an IP address. This option does not ensure that a name server can be reached at the node specified; it simply changes a local variable storing the address of the default name server.

name
Specifies the name of the server.

root
Changes the current server address to the address of the root server. The root server is `ns.nic.ddn.mil` by default, but can be changed using the root=*name* SET subcommand. This command is equivalent to lserver *name*.

An error occurs if the name of the root server cannot be mapped to an IP address. This option does not ensure that a name server can be reached at the node specified; it simply changes a local variable storing the address of the default name server.

server_address
Specifies the IP address of the name server to be queried other than the default name server. A query for the address in the `in-addr.arpa` domain is initially made to the default name server to map the IP address to a domain name for the server.

server_name
Directs the default name server to map *server_name* to an IP address and then use the name server at that IP address.

server *parms*
Changes the current server. If *name* is specified, the IP address of *name* is determined using the current server.

An error occurs if the domain name cannot be mapped to an IP address. This option does not ensure that a name server can be reached at the address; it simply changes a local variable storing the address of the default name server.

set *option*
Changes internal state information values. See “NSLOOKUP options” on page 897 for a description of the options.

view *data_set_name*
Sorts and lists the contents of *data_set_name* one screen at a time. An error occurs if the data set does not exist.

Usage

- You can query by entering the domain name of the node or subnetwork for which information is required. Define the data type of information to be retrieved using the SET *querytype*= option. You can define only one type of resource record for a domain name in a single query, unless the wildcard query type of ANY has been set. If an IP address is given instead of a domain name, a query for the address in the `in-addr.arpa` domain is made to map the IP address to a domain name.

  The domain name or address for the query can be followed by the domain name or IP address of a name server to contact for the query. If this is not specified, the current name server is used. For example, entering:

```
toolah wurrup.fourex.oz
```

  queries the name server on `wurrup.fourex.oz` for information about the node `toolah`. When specifying domain names that include periods, the trailing period (indicating a fully qualified domain name) is optional. NSLOOKUP deletes the trailing period if it is present. If you are specifying a root domain, the domain name must have two trailing periods. For example, specify mynode.. when the node mynode is in the root domain.

- The name server often requires a fully qualified domain name for queries. However, NSLOOKUP enables the specification of a default subnetwork domain using the SET *domain*= option, with the initial default obtained from the TCPIP.DATA data set. When the defname flag is enabled using the SET *defname* option, the default domain name specified by SET *domain*= is appended to all unqualified domain names. For example, if the default domain name is fourex.oz and the *defname* flag is enabled, a query for the name toolah automatically generates a query packet containing the domain name toolah.fourex.oz.
- A timeout error occurs if the name server is not running or is unreachable. A Non-existent Domain error occurs if any resource record type for the specified domain name is not available at the name server. A Server Failed error occurs when the local name server cannot communicate with the remote name server.
- NSLOOKUP might interpret typing or syntax errors in subcommands as queries. This results in a query being sent and the name server response printed. The response is usually Non-existent Domain, which indicates that the server could not find a match for the query.

NSLOOKUP options

The configuration options of NSLOOKUP determine the operation and results of your name server queries. These options can be specified in command-mode queries, interactive-mode queries, or in the *user_id*.NSLOOKUP.ENV data set. When you include NSLOOKUP options with the initial NSLOOKUP command the (**-**) operand must immediately precede the option. If you specify NSLOOKUP options while in interactive mode, the SET subcommand must precede the option. Specifying NSLOOKUP options in the *user_id*.NSLOOKUP.ENV data set is optional. Use the SET subcommand before the option if you want to reset the option value. The (**-**) operand is not valid preceding *options* in the *user_id*.NSLOOKUP.ENV data set.

For example, to specify a name server (NS) type record lookup for the domain name fourex.oz in command mode you enter:

```
nslookup -querytype=ns fourex.oz
```

To submit the same request using interactive mode enter the following sequence:

```
nslookup
set querytype=ns
fourex.oz
```

To make querytype of NS a default option for your NSLOOKUP commands, place one of the following statements in the *user_id*.NSLOOKUP.ENV data set:

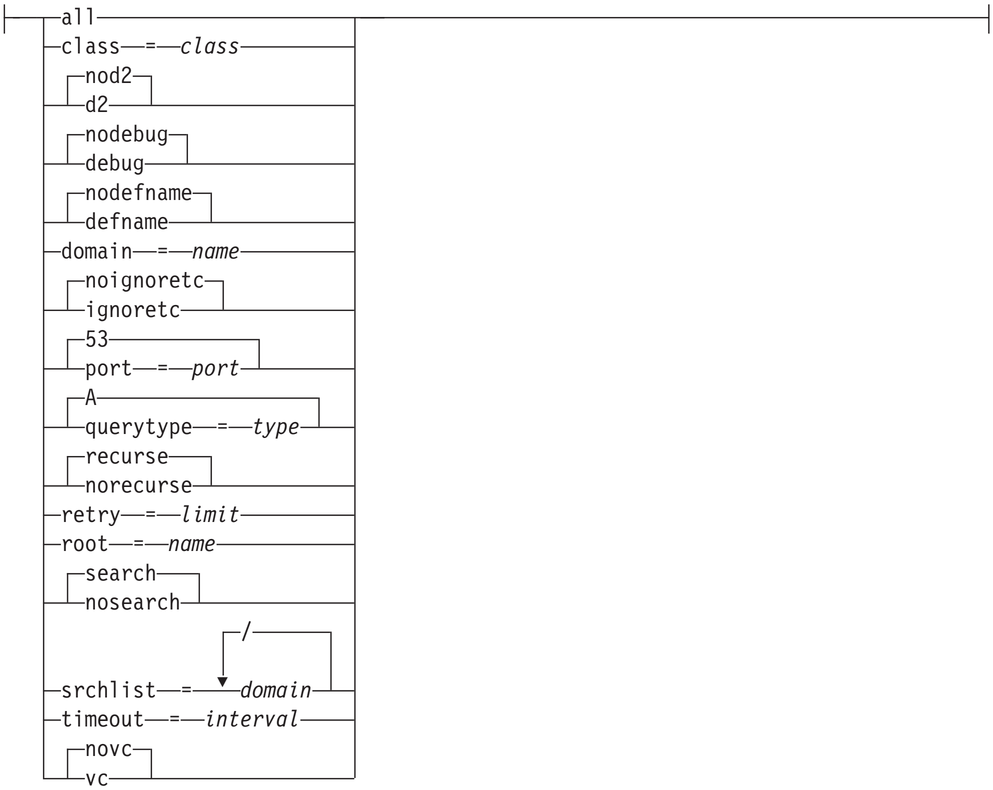
- *set querytype=ns*
- *querytype=ns*

The optional data set *user_id*.NSLOOKUP.ENV. contains only NSLOOKUP options and defines the NSLOOKUP defaults. If the *user_id*.NSLOOKUP.ENV data set exists, the NSLOOKUP options are read from the data set and executed before any queries are made. You must enter each option on a separate line. Blank lines are ignored.

The following example shows the contents of the *user_id*.NSLOOKUP.ENV data set:

```
set domain=powers.oz
querytype=HINFO
set norecurse
vc
```

Option:



all
: Enables you to print the current values of the internal state variables. This option does not alter the internal state of NSLOOKUP.

class=*class*
: Sets the class of information returned by queries. The class must be identified by its mnemonic. The minimum abbreviation for this option is *cl*.

d2
: Directs NSLOOKUP to enable extra debugging mode. Using *d2* also enables debug mode.

  Note: To obtain all alias names for a host when using reverse query, you must set the *d2* option.

nod2
: Directs NSLOOKUP to disable extra debugging mode. The default is *nod2*.

debug
: Directs NSLOOKUP to print debugging information for each query and its corresponding response. The minimum abbreviation is *deb* and *nodeb*.

nodebug
: Directs NSLOOKUP to not print debugging information for each query and its corresponding response. This option also disables *d2*. The minimum abbreviation is *nodeb*. This is the default.

defname

Directs NSLOOKUP to append the default domain name to an unqualified domain name in a query. This value is the default. If the search list is also enabled, then the domain names specified in the search list will also be appended until a query is resolved or until no more domain names are left in the search list.

The default domain name is initially obtained from the TCPIP.DATA data set, but can be changed using the domain=*name* option. The minimum abbreviation for this option is *def*.

nodefname

Directs NSLOOKUP to not append the default domain name to an unqualified domain name in a query.

If you specify this option, the domain name specified in the query is passed to the server without modification. The search list will not be used, even if it is enabled. The minimum abbreviation for this option is *nodef*.

domain=*name*

Sets the default domain name to *name*. Initially, the default domain name is obtained from the TCPIP.DATA data set. The validity of *name* is not verified. This option also updates the search list. The search list contains the domain specified and the parents of the default domain if it has at least two components in its name. For example, if the default domain is `wurrup.forex.oz`, the search list contains `wurrup.forex.oz` and `forex.oz`. Use the SET *srchlist* command to specify a different search list. The minimum abbreviation for this option is *do*.

ignoretc

Directs NSLOOKUP on the handling of truncated responses. The name server indicates, in the response header, that the complete query response did not fit into a single UDP packet and has been truncated.

Specifying *ignoretc* directs NSLOOKUP to ignore the truncation condition when it is set in the response by the name server.

NSLOOKUP does not handle responses greater than 512 characters in length. Responses greater than 512 characters are truncated and the internal truncation flag is set. This condition is revealed only when the *debug* option is enabled. The minimum abbreviation for this option is *ig*.

noignorectc

Directs NSLOOKUP to automatically retry the query using a TCP connection when a response is sent with the truncation indicator set. This is the default. The minimum abbreviation for this option is *noig*.

port=*port*

Specifies the port number to use when contacting the name server. The Domain Name System is a well-known service and has been allocated port 53. NSLOOKUP uses port 53 by default, but the port option enables you to specify another port to access. The minimum abbreviation for this option is *po*.

querytype=*type*

Specifies the type of information returned by queries. The initial query type is A (address information). See the z/OS Communications Server: IP Configuration Reference for detailed information about available query types.

NSLOOKUP cannot generate queries about type NULL. However, it can accept responses containing resource records of type NULL. In this case, NSLOOKUP displays the number of bytes returned in the NULL record. Global queries that return all resource records for a specific domain name are specified by the wildcard value ANY. The minimum abbreviation for this option is *q*.

The type=*type* option is accepted by NSLOOKUP as a synonym for the querytype=*type* option.

recurse
Directs NSLOOKUP to request a recursive query when querying a name server. The minimum abbreviation for this option is *rec*. This is the default.

norecurse
Specifies that a recursive query is not returned. The minimum abbreviation for this option is *norec*.

retry=*limit*
Specifies the number of times a request is resent. When a request is sent and the timeout period expires for a response, the request is resent until the value specified in *limit* has been exceeded. The value specified in *limit* determines the number of attempts made to contact the name server. The default value for *limit* is retrieved from the TCPIP.DATA data set.

Setting *limit* to 0 disables NSLOOKUP from contacting the name server. The result is an error message `no response from server`.

The retry algorithm for NSLOOKUP uses both the *limit* value and the timeout period. The minimum abbreviation for this option is *ret*.

root=*name*
Specifies the name of a root server. The root server is `ns.nic.ddn.mil` by default.

search
Directs NSLOOKUP to enable the use of a search list. The minimum abbreviation for this option is *sea*.

nosearch
Directs NSLOOKUP to disable the use of a search list. The minimum abbreviation for this option is *nosea*.

srchlist=[*domain***/***domain***/...]**
Specifies one or up to three domain names to be appended to non-fully-qualified domain names when attempting to resolve the host name. Each domain name specified is tried in turn until a match is found or no domain names are left to try.

This option also directs the default domain to be set to the first domain name specified in the search list. The minimum abbreviation for this option is *srchl*.

timeout=*interval*
Specifies the number of seconds to wait before timing out of a request. The default for *interval* is retrieved from the TCPIP.DATA data set. The minimum abbreviation for this option is *t*.

vc Specifies to use a virtual circuit (TCP connection) to transport queries to the name server or datagrams (UDP). The default is retrieved from the TCPIP.DATA data set.

novc
Specifies to not use a virtual circuit to transport queries to the name server or datagrams. This option is the default.

NSLOOKUP examples

This material contains examples of NSLOOKUP command-mode queries, and interactive-mode queries using the various options available for NSLOOKUP commands.

In Figure 2, the router wurrup has two IP addresses and there are two name servers, wurrup being the primary name server. This network is described by a single zone in the domain naming hierarchy stored in the name servers.

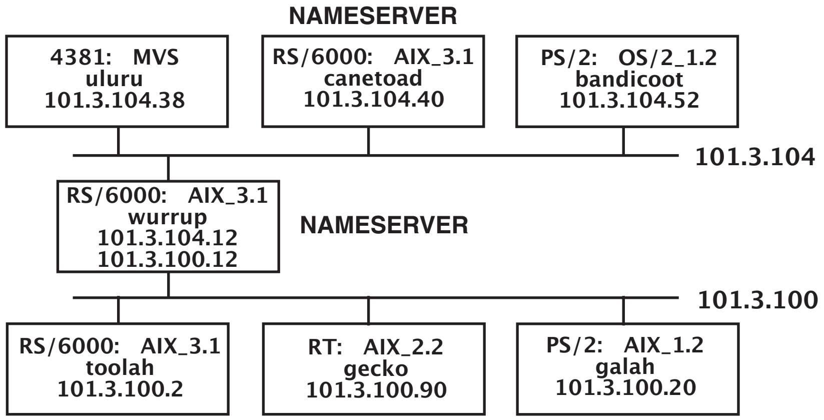


Figure 2. Hierarchical naming tree—A TCP/IP network

The following examples show how to use NSLOOKUP to extract information from a name server. The queries are executed from the z/OS host uluru on the network described in Figure 2.

The following examples are command-mode queries.

- To make a simple address query:

```
  User:    nslookup toolah.fourex.oz wurrup.fourex.oz
System:  Server:  wurrup
         Address:   101.3.104.12

         Name:   toolah.fourex.oz
         Address:   101.3.100.2
```

- To specify a name server (NS) type record lookup:

```
  User:    nslookup -querytype=ns fourex.oz
System:  Server:  canetoad
         Address:   101.3.104.40

         fourex.oz  nameserver = wurrup.fourex.oz
         fourex.oz  nameserver = canetoad.fourex.oz
         wurrup.fourex.oz   internet address = 101.3.100.12
         wurrup.fourex.oz   internet address = 101.3.104.12
         canetoad.fourex.oz internet address = 101.3.104.40
```

- To specify a different default domain name to be appended to an unqualified domain name given as input:

```
  User:    nslookup -do=fourex.oz uluru
System:  Server:  canetoad.fourex.oz
         Address:  101.3.104.40

         Name:    uluru.fourex.oz
         Address:  101.3.104.38
```

- To specify a list of domain names to be appended in turn to the unqualified host name when attempting to resolve it:

```
  User:    nslookup -srchlist=nowhere.oz/fourex.oz uluru
System:  Server:  canetoad.fourex.oz
         Address:  101.3.104.40

         Name:    uluru.fourex.oz
         Address:  101.3.104.38
```

The following command places NSLOOKUP in interactive mode with wurrup as the default server.

```
User:
nslookup - wurrup
System:   Default Server:  wurrup
Address:   101.3.104.12
```

All following examples are in the interactive mode initiated in the preceding example.

- Show the default flag settings:

```
User:
set all
Default Server:  wurrup.fourex.oz
Address:  101.3.104.12

 Set options:
   nodebug             defname         nosearch        recurse
   nod2                novc            noignoretc      port=53
   querytype=A         class=IN        timeout=60      retry=1
   root=ns.nic.ddn.mil.
   domain=FOUREX.OZ
   srchlist=FOUREX.OZ
```

- Perform a simple address query:

```
User:
toolah
System:    Server:  wurrup
Address:   101.3.104.12

Name:    toolah.FOUREX.OZ
Address:  101.3.100.2
```

- Set the query record type to HINFO, and perform another query:

```
User:
set q=HINFO
toolah
System:     Server:  wurrup
Address:   101.3.104.12

toolah.FOUREX.OZ   CPU = RS6000     OS = AIX3.1
```

- Find out the name servers available for a domain:

```
User:
set q=NS
fourex.oz
System:   Server:  wurrup
Address:   101.3.104.12

fourex.oz  nameserver = wurrup.fourex.oz
fourex.oz  nameserver = canetoad.fourex.oz
wurrup.fourex.oz   internet address = 101.3.100.12
wurrup.fourex.oz   internet address = 101.3.104.12
canetoad.fourex.oz internet address = 101.3.104.40
```

- Change the current server from wurrup to canetoad and make more queries:

```
User:   server canetoad
System:   Default Server:  canetoad.FOUREX.OZ
Address:   101.3.104.40

User:
set q=A
gecko
System:   Server:  canetoad.FOUREX.OZ
Address:   101.3.104.40

Name:    gecko.FOUREX.OZ
Address:  101.3.100.90
```

- Enable debugging and execute a simple query to see the result, and then disable debugging:

```
User:
set deb
wurrup
System:   Server:  canetoad.FOUREX.OZ
Address:   101.3.104.40

          res_mkquery(0, wurrup.FOUREX.OZ, 1, 1)
          ------------
          Got answer:
             HEADER:
                opcode = QUERY, id = 7, rcode = NOERROR
                header flags:  response, auth. answer, want recursion,
                recursion avail
                questions = 1, answers = 2, authority records = 0,
                additional = 0

             QUESTIONS:
                wurrup.FOUREX.OZ, type = A, class = IN
             ANSWERS:
             -> wurrup.FOUREX.OZ
                internet address = 101.3.104.12
                ttl = 9999999 (115 days 17 hours 46 mins 39 secs)
             -> wurrup.FOUREX.OZ
                internet address = 101.3.100.12
                ttl = 9999999 (115 days 17 hours 46 mins 39 secs)

          ------------
Name:    wurrup.FOUREX.OZ
         Addresses:  101.3.104.12, 101.3.100.12
User:
set nodeb
```

- Find all addresses in the `fourex.oz` domain using the *ls* option:

```
User:
ls fourex.oz
System:   [canetoad.FOUREX.OZ]
          fourex.oz                     server = wurrup.fourex.oz
          wurrup                         101.3.100.12
          wurrup                         101.3.104.12
          fourex.oz                     server = canetoad.fourex.oz
          canetoad                       101.3.104.40
          gecko                          101.3.100.90
          wurrup                         101.3.100.12
          wurrup                         101.3.104.12
          galah                          101.3.100.20
          bandicoot                      101.3.104.52
          toolah                         101.3.100.2
          canetoad                       101.3.104.40
          loopback                       127.0.0.1
          uluru                          101.3.104.38
```

- Find all aliases in the `fourex.oz` domain, then exit from NSLOOKUP interactive mode:

```
User:
ls -a fourex.oz
System:  [canetoad.FOUREX.OZ]
          localhost                      loopback.fourex.oz
          infoserver                     wurrup.fourex.oz
          pabxserver                     wurrup.fourex.oz
User:
exit
```

- To display a summary of available commands:

```
User:
help
System:
Commands:       (identifiers are shown in uppercase, <> means optional)
NAME            - print info about the host/domain NAME using default server
NAME1 NAME2     - as above, but use NAME2 as server
help or ?     - print info on common commands; see nslookup man for details
set OPTION      - set an option
    all         - print options, current server and host
    <no>debug   - print debugging information
    <no>d2      - print exhaustive debugging information
    <no>defname - append domain name to each query
    <no>recurse - ask for recursive answer to query
    <no>vc      - always use a virtual circuit
    domain=NAME - set default domain name to NAME
    srchlist=N1</N2/.../N6> - set domain to N1 and search list to N1,N2, etc.
    root=NAME   - set root server to NAME
    retry=X     - set number of retries to X
    timeout=X   - set initial time-out interval to X seconds
    querytype=X - set query type, e.g., A,ANY,CNAME,HINFO,MX,NS,PTR,SOA,WKS
    type=X      - synonym for querytype
    class=X     - set query class to one of IN (Internet), CHAOS, HESIOD or ANY
server NAME     - set default server to NAME, using current default server
lserver NAME    - set default server to NAME, using initial server
finger <USER>   - finger the optional NAME at the current default host
root            - set current default server to the root
ls <opt> DOMAIN ^> DATASET| - list addresses in DOMAIN
                  (optional: output to DATASET)
    -a          -  list canonical names and aliases
    -h          -  list HINFO (CPU type and operating system)
    -s          -  list well-known services
    -d          -  list all records
    -t TYPE     -  list records of the given type (e.g., A,CNAME,MX, etc.)
view DATASET  - sort an 'ls' output file and view it with more
exit          - exit the program
```

- To find information for all the users currently logged in on the node specified in the last address query:

```
User:
finger
System:
[canetoad.FOUREX.OZ]
Further output to be generated ....
```

- To set the default domain name to `fourex.oz`, use the command

  ```
  set domain=fourex.oz
  ```

  This command overrides the DOMAINORIGIN statement in the *tcpip*.TCPIP.DATA data set.
- To specify that the default domain name is to be appended to an unqualified domain name given in a query, use the SET *defname* command.
- To request that the query be resent three times if the timeout period expires for a response, use the SET *retry=3* command. A value of 3 is the maximum valid value.

Using the z/OS UNIX onslookup/nslookup command

The z/OS UNIX **nslookup** is a program used to query Internet domain name servers. The **nslookup** command has two modes: interactive and non-interactive. Use the interactive mode to query name servers for information about various hosts and domains. Non-interactive mode is used to display just the name and requested information for a host or domain.

Rule: The **onslookup**/**nslookup** command does not use resolver caching.

The z/OS UNIX **onslookup**/**nslookup** command enables you to perform the following tasks from the z/OS UNIX environment:
- Identify the location of name servers
- Examine the contents of a name server database
- Establish the accessibility of name servers

See "nslookup versions" on page 906 for listings of valid start options and subcommands for the different versions of **nslookup**.

To display a list of options, enter the following from the command line:

```
onslookup -h
```

Note:
1. The **onslookup** command is a synonym for the **nslookup** command in the z/OS UNIX shell. The **nslookup** command syntax is the same as that for the **onslookup** command. The **nslookup** command can be run from the z/OS UNIX shell or from TSO; however, only the legacy TSO version of NSLOOKUP is available from TSO.
2. The **onslookup** messages are not documented in the z/OS Communications Server library. Therefore, **onslookup** command messages do not give a message ID for debugging.

The **onslookup** command has two modes of operation: interactive mode and command mode. In both modes, the address of the default name server comes from the resolver configuration file.

In the following example, the default domain is raleigh.ibm.com, and the default name server is at 9.37.34.149. If that name server fails to respond, the one at 9.37.34.7 is used.

```
domain   raleigh.ibm.com
nameserver   9.37.34.149
nameserver   9.37.34.7
```

onslookup configuration

BIND 9 DNS uses the z/OS application's search order to find TCPIP.DATA statements. See the z/OS Communications Server: IP Configuration Guide for details. It uses the following directives from the resolver configuration file.

1. `nameserver/nsinteraddr`
2. `options ndots:`*n*
3. `search`
4. `domain/domainorigin`

The value specified by *ResolverTimeout* in the `/etc/resolv.conf` file has priority over the value specified by `ResolverTimeout` in the TCPIP.DATA configuration data set. See "Resolver related commands" on page 891 for detailed descriptions.

See the z/OS Communications Server: IP Configuration Guide for detailed information about **onslookup** configuration.

nslookup versions

This material presents similarities and differences for the two separate versions (TSO and z/OS UNIX shell) of the **nslookup** command. The following tables present the start options and subcommands used by each version. The only version of **nslookup** that supports IPv6 addresses and resource record types is **nslookup** that is invoked from the z/OS UNIX shell.

The following table shows the validity of start options between TSO and z/OS UNIX shell **nslookup** commands.

Table 19. Start option validity for NSLOOKUP TSO and z/OS UNIX shell

Start option	TSO NSLOOKUP	nslookup in the z/OS UNIX shell
-all	Yes	Yes
-class	Yes	Yes
-cl (short for -class)	Yes	Yes
-diffstamp	No	No
-d2	Yes	Yes
-nod2	Yes	Yes
-debug	Yes	Yes
-deb (short for -debug)	Yes	Yes
-nodebug	Yes	Yes
-nodeb (short for -nodebug)	Yes	Yes
-defname	Yes	Yes
-def (short for -defname)	Yes	Yes
-nodefname	Yes	Yes

Table 19. Start option validity for NSLOOKUP TSO and z/OS UNIX shell (continued)

Start option	TSO NSLOOKUP	nslookup in the z/OS UNIX shell
-nodef (short for -nodefname)	Yes	Yes
-domain=	Yes	Yes
-do= (short for -domain=)	Yes	Yes
-help	No	No
-h	No	Yes
-ignoretc	Yes	No
-ig (short for -ignoretc)	Yes	No
-noignoretc	Yes	No
-noig (short for -noignoretc)	Yes	No
-port=	Yes	Yes
-po= (short for -port=)	Yes	Yes
-querytype=	Yes	Yes
-q= (short for -querytype)	Yes	No
-type= (short for -querytype=)	Yes	Yes
-ty= (short for -querytype=)	No	Yes
-query= (short for -querytype=	No	Yes
-qu= (short for -querytype=)	No	Yes
-recurse	Yes	Yes
-rec (short for -recurse)	Yes	Yes
-norecurse	Yes	Yes
-norec (short for -norecurse)	Yes	Yes
-retry=	Yes	Yes
-ret= (short for -retry=)	Yes	Yes
-root=	Yes	No
-search	Yes	Yes
-sea (short for -search)	Yes	Yes
-nosearch	Yes	Yes
-nosea (short for -nosearch)	Yes	Yes
-srchlist=	Yes	No
-srchl= (short for -searchlist=)	Yes	No
-timeout=	Yes	Yes
-t= (short for -timeout=)	Yes	Yes
-tstamp	No	No
-nostamp	No	No
-sil	No	Yes
-vc	Yes	Yes
-novc	Yes	Yes
-V=	No	Yes

onslookup/nslookup (command mode): Querying a name server in command mode

Command (non-interactive) mode is used to print just the name and requested information for a host or domain. Use the command mode entry of **onslookup** command to specify a single query.

Command mode query is invoked when the name or IP address of the host to be looked up is given as the first argument. The optional second argument specifies the host name or address of a name server.

Note:

1. The **nslookup** command is a synonym for the **onslookup** command in the z/OS UNIX shell. **nslookup** command syntax is the same as that for the **onslookup** command.
2. The **onslookup help** command works only in the interactive mode.

Format

```
►►──onslookup──┬──────────────────┬──┬─name────┬──┬──────────────────┬──────────────►◄
               │  ┌────────────┐  │  └─address─┘  ├─server_name──────┤
               │  ▼            │  │               └─server_address───┘
               └──┤ -Option ├──┴──┘
```

Parameters

-Option
For a description of the **onslookup** options, see "onslookup options" on page 912.

name
Queries the name server for the current query-type of name. The name typically represents a host name.

address
Reverses the components of the address and generates a pointer type (PTR) query to the name server for the in-addr.arpa domain mapping of the address to a domain name.

server_name
Directs the default name server to map *server_name* to an IP address and then uses the name server at that address. This argument is optional. The default is the default name server found by the search order described in "onslookup configuration" on page 906. This name can be a name that resolves to an IPV4 or an IPv6 address.

server_address
Specifies the IP address of the name server to be queried other than the default name server. A query for the address is initially made to the default name server to map the IP address to a domain name for the server. This argument is optional. The default value is the default name server found by the search order described in "onslookup configuration" on page 906. This address can be an IPv4 or an IPv6 address.

Usage

Parameter values and domain names are not case sensitive.

Context

Options can also be specified on the command line if they precede the arguments and are prefixed with a hyphen. For example, to change the default query type to host information and the initial timeout to 10 seconds, type:

```
nslookup -query=hinfo  -timeout=10
```

To display a list of options, enter the following from the command line:

```
onslookup -h
```

For a complete list and description of **onslookup** options, see "onslookup options" on page 912.

onslookup/nslookup (interactive mode): Issuing multiple queries to name servers

Interactive mode enables you to query one or more name servers repeatedly for information about various hosts and domains, to display that information on your console, and, in some cases, to write response data to a file. **nslookup** is a synonym for the **onslookup** command in the z/OS UNIX shell. **nslookup** command syntax is the same as that for the **onslookup** command.

You can enter the interactive mode under the following conditions only:

- No arguments are supplied on command invocation. The default name server is used.
- The first argument is a hyphen (-), and the second argument is the host name or IP address of a name server.

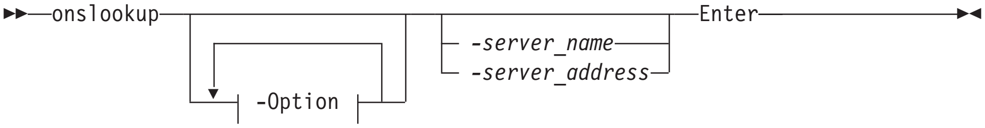
In interactive mode:

- An initial query is made to the selected name server to verify that the server is accessible.
- All subsequent interactive queries are sent to that server unless you specify another server using the *server* or *lserver* subcommands.
- The command line length must be less than 256 characters.
- To treat a built-in command as a host name, precede it with an escape character (\). An unrecognized command is interpreted as a host name.

For a complete list and description of **onslookup** options, see "onslookup options" on page 912. See "nslookup versions" on page 906 for listing of valid commands and start options.

Format

- Interactive mode



- Interactive commands, subsequent queries

```
►►──┬──────────────────────────┬──Enter──────────────────────────────►◄
    ├─exit─────────────────────┤
    ├─host──┬────────────┬─────┤
    │       └─server─────┘     │
    ├─lserver──┬─name──────┬───┤
    │          └─address───┘   │
    ├─server──┬─name──────┬────┤
    │         └─address───┘    │
    └─set──┤ option ├──────────┘
```

Parameters

-Option
For a description of the **onslookup** options, see "onslookup options" on page 912.

-server_name
Directs the default name server to map *-server_name* to an IP address and then uses the name server at that address. This argument is optional. The default is the default name server found by the search order described in "onslookup configuration" on page 906. This name can be a name that resolves to an IPv4 or an IPv6 address.

-server_address
Specifies the IP address of the name server to be queried other than the default name server. A query for the address is initially made to the default name server to map the IP address to a domain name for the server. This argument is optional. The default is the default name server found by the search order described in "onslookup configuration" on page 906. This can be an IPv4 or an IPv6 address.

exit
Exits from **onslookup** interactive mode.

host
host is the host name or IP address you want the name server to resolve. Use this format to look up information for a host using the current default server or using *server* if specified. If **host** is an IP address and the query type is A or PTR, the name of the host is returned. If **host** is a name and does not have a trailing period, the default domain name is appended to the name. (This behavior depends on the state of the set options -domain, -srchlist, -defname, and -search.) To look up a host not in the current domain, append a period to the name.

lserver
Change the default server to one determined by *name* or *address*. This command uses the initial server to look up information about the new server. This can be a name that resolves to an IPv4 or an IPv6 address, or an actual IPv4 or IPv6 address.

If an authoritative answer cannot be found, the names of servers that might have the answer are returned.

server
Change the default server to one determined by *name* or *address*. This command uses the current default server to look up information about the new server. This can be a name that resolves to an IPv4 or IPv6 address, or an actual IPv4 or IPv6 address.

If an authoritative answer cannot be found, the names of servers that might have the answer are returned.

set *keyword*
Allows changes to query environment. The following describes the *keyword* and values that can be used.

all
Prints the current values of the frequently used options to set. Information about the current default server and host is also printed.

class=*query_class*
The class specifies the protocol group of the information. The `class` changes the query class. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query types. The default class is **IN**. The keyword `class` can be abbreviated as `cl`.

[no]d2
Turn exhaustive debugging mode on (`d2`) or off (`nod2`). You will not see any difference between debug, d2 and trace resolver. This turns on **nslookup** internal trace. The default is `nod2`.

[no]debug
Turn basic debugging mode on (`debug`) or off (`nodebug`). Information is printed about the packet sent to the server and the resulting answer. The default is `nodebug`. The keyword `debug` can be abbreviated as `deb`.

[no]defname
If set, append the default domain name to a single-component lookup request (one that does not contain a period). The default is `defname`. The keyword `defname` can be abbreviated as `def`. This is equivalent to the [no]search option. Specifying the '-domain=' option also causes the default domain name to be appended to the name being queried.

domain=*name*
Change the default domain name to *name*. The default domain name is appended to a lookup request depending on the state of the defname and search options. The keyword `domain` can be abbreviated as `do`. This also causes the -search option to be turned on. Also, this becomes the default search list and overrides the default domain specified in the resolver configuration file.

port=*value*
Change the default TCP/UDP name server port to *value*. The default port number is 53. The keyword `port` can be abbreviated as `po`.

querytype=*type*

- The keyword `querytype` can be abbreviated as `type`.
- The keyword `querytype` can also be abbreviated as `qu`.

Change the type of information query *type*. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query types. The default *value* is A.

[no]recurse
Tell the name server to query other servers if it does not have the information. The default is `recurse`. The keyword `recurse` can be abbreviated as `rec`.

retry=*number*
Set the number of retries to *number*. When a reply to a request is not

received within a certain amount of time (changed with `set timeout`), the timeout period is doubled and the request is resent. The retry value controls how many times a request is sent before giving up. The default is 4. The keyword `retry` can be abbreviated as `ret`.

[no]search
If the lookup request contains at least one period but does not end with a trailing period, append the domain names in the domain search list to the request until an answer is received. The default is `search`. The keyword `search` can be abbreviated as `sea`. This is equivalent to the [no]defname option.

timeout=*number*
Change the initial timeout interval for waiting for a reply to *number* seconds. Each retry doubles the timeout period. The default is 5 seconds. The kewyord `timeout` can be abbreviated as `t`.

[no]vc
Always use a virtual circuit (TCP) when sending requests to the server. The default is `novc`. The keyword `vc` can be abbreviated as `v`.

onslookup options

The configuration options of **onslookup** determine the operation and results of name server queries. These options can be specified in command-mode queries, interactive-mode queries, or by the methods described in “onslookup configuration” on page 906.

When you include **onslookup** options with the initial **onslookup** command, the hyphen (-) operand must immediately precede the option. If you specify **onslookup** options while in interactive mode, the SET subcommand must precede the option.

For example, to specify a name server (NS) type record lookup for the domain name `fourex.oz` in command mode you enter:

```
onslookup -querytype=ns fourex.oz
```

To submit the same request using interactive mode, enter the following sequence:

```
onslookup
set querytype=ns
fourex.oz
```

Format

```
├──┬──────────────────────────────────────┬────────────────────────────────────────┤
   ├─all──────────────────────────────────┤
   │                ┌─IN─────┐            │
   ├─class──=───────┼─ANY────┼────────────┤
   │                ├─CHAOS──┤            │
   │                └─HESIOD─┘            │
   │  ┌─nod2─┐                            │
   ├──┴─d2───┴────────────────────────────┤
   │  ┌─nodebug─┐                         │
   ├──┴─debug───┴─────────────────────────┤
   │  ┌─defname───┐                       │
   ├──┴─nodefname─┴───────────────────────┤
   ├─domain──=──name──────────────────────┤
   ├─h────────────────────────────────────┤
   │          ┌─53───────────┐            │
   ├─port─────┴─=port_number─┴────────────┤
   │               ┌─A──────────────────────┐
   ├─querytype─────┴─=resource_record_type──┴┤
   │  ┌─recurse───┐                        │
   ├──┴─norecurse─┴───────────────────────┤
   ├─retry──=──limit──────────────────────┤
   │  ┌─search───┐                         │
   ├──┴─nosearch─┴────────────────────────┤
   ├─sil──────────────────────────────────┤
   ├─timeout──=──interval─────────────────┤
   └─V──=──┬──────┬───────────────────────┘
           └─v9───┘
```

Parameters

-all
Prints the current values of the frequently used options to set. Information about the current default server and host is also printed.

-class=*query_class*
The class specifies the protocol group of the information. Changes the class to *query class*. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid classes. The default class is **IN**. The option `class` can be abbreviated as `cl`.

-[no]d2
Turn exhaustive debugging mode on (`d2`) or off (`nod2`). You will not see any difference between debug, d2, and trace resolver. This turns on **nslookup** internal trace. The default is `nod2`.

-[no]debug
Turn basic debugging mode on (`debug`) or off (`nodebug`). Information is printed about the packet sent to the server and the resulting answer. The default is `nodebug`. The option `debug` can be abbreviated as `deb`.

-[no]defname
If set, append the default domain name to a single-component lookup request (one that does not contain a period). The default is `defname`. The option `defname` can be abbreviated as `def`.

-domain=*name*
The default domain name is appended to a lookup request depending on the state of the defname and search options. The domain search list contains the parents of the default domain if it has at least two components in its name. For example, if the default domain is CC.Berkeley.EDU, the search list is

CC.Berkeley.EDU and Berkeley.EDU. Use the `set srchlist` command to specify a different list. Use the `set all` command to display the list. The option `domain` can be abbreviated as `do`.

-h Prints a brief summary of commands.

-port=*port_number*
Change the default TCP/UDP name server port to *port_number*. The default port number is 53. The option `port` can be abbreviated as `po`.

-querytype=*resource_record_type*

- The option `querytype` can be abbreviated as `type`.
- The option `querytype` can also be abbreviated as `qu`.

Change the type of information query *resource_record_type*. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query types. The default *resource_record_type* is A.

-[no]recurse
Tell the name server to query other servers if it does not have the information. The default is `recurse`. The option `recurse` can be abbreviated as `rec`.

-retry=*limit*
Set the number of retries to *limit*. When a reply to a request is not received within a certain amount of time (changed with the command `set timeout`), the timeout period is doubled and the request is resent. The retry value controls how many times a request is sent before giving up. The default is 4. The option `retry` can be abbreviated as `ret`.

-[no]search
If the lookup request contains at least one period but does not end with a trailing period, append the domain names in the domain search list to the request until an answer is received. The default is `search`. The option `search` can be abbreviated as `sea`.

-sil
Suppress deprecation message.

-timeout=*interval*
Change the initial timeout interval for waiting for a reply to *interval* seconds. Each retry doubles the timeout period. The default is 5 seconds. The keyword `timeout` can be abbreviated as `t`.

-V Sets BIND 9 mode.

-[no]vc
Always use a virtual circuit (TCP) when sending requests to the server. The default is `novc`. The option `vc` can be abbreviated as `v`.

onslookup: diagnosing problems

The **onslookup** program lets you query other name servers with the same query packet another name server would use. This is helpful in diagnosing lookup problems in TCP/IP UNIX System Services.

To turn debugging on at level 1, enter the following commands from the z/OS shell:

```
onslookup
set debug
```

The **onslookup** program shows timeouts and displays response packets.

To turn the debug option off, enter the following command:

```
set nodebug
```

You can set the debugging option to level 2 by entering the `set d2` command just as the `set debug` command was entered previously. The `set d2` command provides program trace information and `set debug` shows parts of the formatted DNS response message.

The resolver shows the normal debugging information plus the query packets that were sent out. Turning on *d2* also turns on *debug*. Turning off *d2*, however, turns off only *d2*; *debug* remains on. To turn off both *d2* and *debug*, enter the command `set` *`nodebug`*.

If the lookup request was not successful, an error message is printed. Possible errors include:

Timed out
The server did not respond to a request after a certain amount of time (changed with `set timeout=`*`value`*) and a certain number of retries (changed with `set retry=`*`value`*).

No response from server
No name server is running on the server machine.

No records
The server does not have resource records of the current query type for the host, although the host name is valid. The query type is specified with the `set querytype` command.

Non-existent domain
The host or domain name does not exist.

Connection refused
The host or domain name refused the connection.

Network is unreachable
The connection to the name could not be made at the current time. This error commonly occurs with ls requests.

Server failure
The name server found an internal inconsistency in its database and could not return a valid answer.

Refused
The name server refused to service the request.

Format error
The name server found that the request packet was not in the correct format. It might indicate an error in **`nslookup`**.

Note: The **`onslookup`** messages are not documented in the z/OS Communications Server library. Therefore, **`onslookup`** command messages do not give a message ID for debugging.

For help with **`onslookup`** commands from the command line, type `onslookup -h`.

Using the z/OS UNIX nsupdate command

You can use the **nsupdate** command to create and execute DNS update operations on a host record as defined in RFC 2136 (for DNS 9) to a name server. This allows resource records to be added or removed from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

Rules:

- Do not manually edit zones that are under dynamic control by **nsupdate** or a DHCP server. Manual edits could conflict with dynamic updates and cause data to be lost.
- Do not use the **nsupdate** command to update DNS zones that are managed by the automated domain name registration (ADNR) application. See information about updates to an ADNR-managed zone in the z/OS Communications Server: IP Configuration Guide for more details.

The **nsupdate** command can be used for both IPv4 and IPv6 connections.

The resource records for **nsupdate** using BIND 9 that are dynamically added or removed with **nsupdate** have to be in the same zone. Requests are sent to the zone's master server. This is identified by the MNAME field of the zone's SOA record.

Batch mode is supported when **nsupdate** subcommands are stacked in a file, and the name of the file is specified as the last argument on the command line:

```
nsupdate /tmp/update.zone
```

The file name must not immediately follow the **-d** option.

BIND 9 DNS uses the z/OS application's search order to find TCPIP.DATA statements. See the z/OS Communications Server: IP Configuration Guide for details. It uses the following directives from the resolver configuration file:

1. nameserver/nsinteraddr
2. options ndots:n
3. search domain/domainorigin

nsupdate: Command mode

Use **nsupdate** to create and execute DNS update operations on a host record to a name server. You can add or remove resource records from a zone without manually editing the zone file. A single update request can contain requests to add or remove more than one resource record.

You can use this command in an interactive fashion (where you are prompted through a series of subcommands and associated input values), or if you know the sequence of operations and input values beforehand, you can use **nsupdate** in batch mode. You can read input from a file. The file name must appear at the end of the **nsupdate** command line and must not follow the **-d** option.

Format

```
                 ┌─────────────────────────────┐
                 ▼                             │
►►──nsupdate─────┬─────────────────────────────┬┴──┬─────────────────┬──────────────►◄
                 ├─ -d─────────────────────────┤   └─batch_file_name─┘
                 ├─ -v─────────────────────────┤
                 ├──┬─ -y ──keyname:secret─┬───┤
                 │  └─ -k ──keyfile────────┘   │
                 ├─ -D─────────────────────────┤
                 ├─ -M─────────────────────────┤
                 └─ -V──┬─────┬────────────────┘
                        └─ v9─┘
```

Parameters

batch_file_name
The name of a z/OS UNIX file that contains **nsupdate** subcommands, which can be used as input to the **nsupdate** command. If the *batch_file_name* does not specify a directory, the file must be in the current directory. The file name can contain v9 **nsupdate** commands, one per line.

-d Turn debug trace on. This provides tracing information about the update requests that are made and the replies received from the name server. Use this option if you want to see the response from the server on the **nsupdate** client side.

-v By default **nsupdate** uses UDP to send update requests to the name server. The **-v** option makes **nsupdate** use a TCP connection. This might be preferable when a batch of update requests is made.

-y *keyname:secret*
nsupdate uses the **-y** or **-k** option to provide the shared-secret needed to generate a TSIG record for authenticating Dynamic DNS update requests. These options are mutually exclusive. When the **-y** option is used, a signature is generated from *keyname:secret*. The name of the key is *keyname*, and *secret* is the base-64 encoded shared-secret. Use of the **-y** option is discouraged because the shared-secret is supplied as a command line argument in clear text. This might be visible in the output from `ps -ef` or in a history file maintained by the user's shell.

-k *keyfile*
nsupdate uses the **-y** or **-k** option to provide the shared-secret needed to generate a TSIG record for authenticating Dynamic DNS update requests. These options are mutually exclusive. With the **-k** option, **nsupdate** reads the shared-secret from the file *keyfile*, whose name is of the form K{name}.+157.+{random}.private. For historical reasons, the file K{name}.+157.+{random}.key must also be present.

-D Turn debug trace and procedure trace on.

-M Turn debug, procedure, and memory trace on.

-V Specifies the version of **nsupdate**. The only valid version is v9.

Transaction signatures can be used to authenticate the Dynamic DNS updates. These use the TSIG resource record type described in RFC 2845. The signatures rely on a shared-secret that should be known only to **nsupdate** and the name server. Currently, the only supported encryption algorithm for TSIG is HMAC-MD5, which is defined in RFC 2104. Suitable key{} statements and allow-update{} or update-policy{} options must be added to the BIND 9 name server configuration file (for example, /etc/named.conf) so that the name server can authorize **nsupdate** clients that use TSIG authentication. **nsupdate** does not read /etc/named.conf.

nsupdate: Subcommand mode

The following subcommands can be used in the **nsupdate** command shell. **nsupdate** reads commands from its standard input. Each command is supplied on exactly one line of input. Some commands are for administrative purposes. The others are either update instructions or prerequisite checks on the contents of the zone.

Format

Start nsupdate subcommand mode

```
►►──nsupdate──Enter────────────────────────────────────────────────────►◄
```

Subsequent subcommand entry

```
      ┌────────────────────────────────┐
      ▼                                │
►►────┬─────────────────────────┬─Enter┴──────────────────────────────►◄
      ├─quit────────────────────┤
      ├─prereq───┬─nxdomain─┬───┤
      │          ├─yxdomain─┤   │
      │          ├─nxrrset──┤   │
      │          └─yxrrset──┘   │
      ├─server──────────────────┤
      ├─send────────────────────┤
      ├─show────────────────────┤
      ├─update───┬─add────┬─────┤
      │          └─delete─┘     │
      └─zone────────────────────┘
```

Parameters

The following subcommands can be used in the **nsupdate** command shell. Some of these subcommands make prerequisite checks on the contents of the zone. These checks set conditions that some name or set of resource records (RRset) either exists or is absent from the zone. These conditions must be met if the entire update request is to succeed. Updates are rejected if the tests for the prerequisite conditions fail.

Every update request consists of 0 or more prerequisites and 0 or more updates. This allows a suitably authenticated update request to proceed if some specified resource records are present or missing from the zone. A blank input line causes the accumulated commands to be sent as one Dynamic DNS update request to the name server.

quit Quits the program.

prereq nxdomain *domain-name*
Requires that no resource record of any type exists with name *domain-name*.

prereq yxdomain *domain-name*
Requires that *domain-name* exists (has as at least one resource record, of any type).

prereq nxrrset *domain-name [class] type*
Requires that no resource record exists of the specified *type*, *class* and *domain-name*. If *class* is omitted, IN (Internet) is assumed. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid classes and types.

prereq yxrrset *domain-name [class] type*
This requires that a resource record of the specified *type*, *class* and *domain-name* must exist. If *class* is omitted, IN (Internet) is assumed. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid classes and types.

prereq yxrrset *domain-name [class] type data...*
The data from each set of prerequisites of this form sharing a common *type*, *class* and *domain-name* are combined to form an RRset. This RRset must exactly match the RRset existing in the zone at the given *type*, *class* and *domain-name*. The data is written in the standard text representation of the resource record's RDATA. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid classes and types.

server *servername* **[***port***]**
Specifies the server name or IP address where all dynamic update requests are sent. This can be an IPv4 or an IPv6 address or a name that resolves to an IPv4 or IPv6 address. When no server statement is provided, nsupdate sends updates to the master server of the correct zone. The latter capability, like use of a server name instead of IP address, is assuming **nsupdate** can find resolver data to connect to a name server.

The MNAME field of that zone's SOA record will identify the master server for that zone. *port* is the port number on *servername* where the dynamic update requests get sent. If no port number is specified, the default DNS port number is 53.

send Sends update to server.

show Shows update to be sent.

update delete *domain-name [class] [type [data...]]*
Deletes any resource records named *domain-name*. If *type* and *data* is provided, only matching resource records are removed. If *class* is omitted, IN (Internet) is assumed.

update add *domain-name ttl [class] type data..*
Adds a new resource record with the specified *ttl*, *class*, *type* and *data*. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid classes and types.

zone *zonename*
Specifies that all updates are to be made to the zone *zonename*. If no zone statement is provided, **nsupdate** will attempt to determine the correct zone to update based on the rest of the input.

nsupdate BIND v9 examples

You can see the following examples of **nsupdate** BIND v9.

How to insert and delete resource records

The examples below show how **nsupdate** with BIND 9 could be used to insert and delete resource records from the example.com zone. Notice that the input in each example contains a trailing blank line so that a group of commands are sent as one dynamic update request to the master name server for example.com.

```
# nsupdate
  > update delete oldhost.example.com A
  > update add newhost.example.com 86400 A 172.16.1.1
  >
```

Any A records for oldhost.example.com are deleted, and an A record for newhost.example.com at IP address 172.16.1.1 is added. The newly added record has a 1-day TTL (86400 seconds).

```
# nsupdate
 > prereq nxdomain nickname.example.com
 > update add nickname.example.com CNAME somehost.example.com
 >
```

The prerequisite condition gets the name server to check that there are no resource records of any type for nickname.example.com. If there are, the update request fails. If this name does not exist, a CNAME for it is added. This ensures that when the CNAME is added, it cannot conflict with the long-standing rule in RFC 1034 that a name must not exist as any other record type if it exists as a CNAME. (The rule has been updated for DNSSEC in RFC 2535 to allow CNAMEs to have SIG, KEY and NXT records.)

How to use an input file for nsupdate

The example below shows how to have **nsupdate** read subcommands from a file for BIND 9.

1. Create a z/OS UNIX file containing the following **nsupdate** subcommands. Assume the file is named nsupdate.commands.

   ```
               update delete oldhost.example.com A
               update add newhost.example.com 86400 A 172.16.1.1
               show
               send
               quit
   ```

2. Then issue the following command from the directory where the file, nsupdate.commands resides.

   ```
   nsupdate nsupdate.commands
   ```

3. Because the `zone` and `server` subcommands were not explicitly issued, the defaults will come from the resolver configuration data set. Assume the following information is coded in the resolver configuration data set.

   ```
               domain     example.com
               nameserver 127.0.0.1
   ```

4. The name server on the local host would be used to look up the location of the `example.com` domain. Once the authoritative name server is located, the updates in the nsupdate.commands file are executed and sent to that name server.
5. The output is sent to stdout. The following information might appear on the z/OS UNIX screen.

   ```
   > nsupdate nsupdate.commands
     Running nsupdate version 9
     Allocated socket 6, type udp
     Outgoing update query:
     ;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
     ;; flags: ; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
     ;; UPDATE SECTION:
     oldhost.example.com.    0       ANY     A
     newhost.example.com.    86400   IN      A       172.16.1.1
   ```

Using the TSO DIG command

DIG is a program for querying Domain Name Servers, which enables you to:

- Exercise name servers
- Gather large volumes of domain name information
- Execute simple domain name queries

Rule: The DIG command does not use resolver caching.

If you have a group of queries to be resolved, you must issue an NSLOOKUP command for each query. Compared to NSLOOKUP, the DIG command provides a larger range of options for controlling queries and screen output.

The TSO DIG command has been deprecated in favor of the z/OS UNIX **`dig`** command. TSO DIG will not understand some of the newer resource record types, including many types of IPv6 data.

DIG internal state information

The internal state information of DIG determines the operation and results of your name server queries. You can configure the internal state information of DIG using the following methods, listed in order of preference:

1. TCP/IP client program configuration data set, TCPIP.DATA
2. DIG startup data set, *user_id*.DIG.ENV
3. Query options on the command line or in a batch data set

The *user_id*.DIG.ENV data set contains a list of query option defaults. This list is initialized from the *user_id*.DIG.ENV data set when DIG is invoked. The default values in *user_id*.DIG.ENV are used for all queries unless overridden by query flags on the command line. The defaults can be reset during a batch run by using the *-envset* flag on a batch data set line.

The *user_id*.DIG.ENV data set is created and updated using the *-envset* option, which writes the current defaults out to the data set after parsing the query options on the command line. The *-envset* option specified on the command line and the existing default values are saved in the *user_id*.DIG.ENV data set as the default environment for future invocations of DIG. The *user_id*.DIG.ENV data set is not reread when the environment is updated during batch queries and the *-envset* flag has no effect on subsequent queries in a batch data set. The *user_id*.DIG.ENV data set is written in nontext format, and cannot be viewed or edited.

DIG command: Query name servers

You can use DIG in command mode, where all options are specified on the invoking command line, or in batch mode, where a group of queries are placed in a data set and executed by a single invocation of DIG. DIG provides a large number of options for controlling queries and screen output, including most of the functions of NSLOOKUP.

You can create a data set for batch mode queries using the `-f` *`data_set`* option. The data set contains complete queries, one per line, that are executed in a single invocation of DIG. The keyword *DIG* is not used when specifying queries in a batch data set. Blank lines are ignored, and lines beginning with a # symbol or a semicolon (;) in the first column are comment lines.

Options specified on the initial command line are in effect for all queries in the batch data set unless explicitly overridden. Several options are provided exclusively for use within batch data sets, giving greater control over DIG operation.

Some internal state information is retrieved from the TCPIP.DATA data set. See the z/OS Communications Server: IP Configuration Guide for more information about the TCPIP.DATA data set.

Format

```
►►──DIG──┬─────────┬──┬─────────────┬──┬───────┬──┬────────┬──┬─────────────┬──────────►
         └─@server─┘  └─domain_name─┘  └─qtype─┘  └─qclass─┘  └─%──comment─┘

   ┌─────────────────────┐  ┌───────────────────┐
   ▼                     │  ▼                   │
►──┬─────────────────────┬──┬───────────────────┬─────────────────────────────────────►◄
   └──┤ +queryoption ├───┘  └──┤ -digoption ├───┘
```

+queryoption:

```
├─┬────────────────────────────┬─────────────────────────────────────────┤
  │ ┌─noaaonly─┐               │
  ├─┴─aaonly───┴───────────────┤
  │ ┌─addit───┐                │
  ├─┴─noaddit─┴────────────────┤
  │ ┌─answer───┐               │
  ├─┴─noanswer─┴───────────────┤
  │ ┌─author───┐               │
  ├─┴─noauthor─┴───────────────┤
  │ ┌─nocl─┐                   │
  ├─┴─cl───┴───────────────────┤
  │ ┌─cmd───┐                  │
  ├─┴─nocmd─┴──────────────────┤
  │ ┌─nod2─┐                   │
  ├─┴─d2───┴───────────────────┤
  │ ┌─debug───┐                │
  ├─┴─nodebug─┴────────────────┤
  │ ┌─defname───┐              │
  ├─┴─nodefname─┴──────────────┤
  ├─domain──=──name────────────┤
  │ ┌─Header───┐               │
  ├─┴─noHeader─┴───────────────┤
  │ ┌─header───┐               │
  ├─┴─noheader─┴───────────────┤
  │ ┌─noignore─┐               │
  ├─┴─ignore───┴───────────────┤
  │ ┌─noko─┐                   │
  ├─┴─ko───┴───────────────────┤
  ├─pfand──=──number───────────┤
  ├─pfdef──────────────────────┤
  ├─pfmin──────────────────────┤
  ├─pfor──=──number────────────┤
  ├─pfset──=──number───────────┤
  │ ┌─noprimary─┐              │
  ├─┴─primary───┴──────────────┤
  │ ┌─noqr─┐                   │
  ├─┴─qr───┴───────────────────┤
  │ ┌─ques───┐                 │
  ├─┴─noques─┴─────────────────┤
  │ ┌─recurse───┐              │
  ├─┴─norecurse─┴──────────────┤
  │ ┌─reply───┐                │
  ├─┴─noreply─┴────────────────┤
  ├─retry──=──limit────────────┤
  │ ┌─nosort─┐                 │
  ├─┴─sort───┴─────────────────┤
  │ ┌─stats───┐                │
  ├─┴─nostats─┴────────────────┤
  ├─timeout──=──time_out_value─┤
  │ ┌─ttlid───┐                │
  ├─┴─nottlid─┴────────────────┤
  │ ┌─novc─┐                   │
  └─┴─vc───┴───────────────────┘
```

-digoption:

```
├──┬──────────────────────────────────────┬──────────────────────────────┤
   ├─c── ─query_class─────────────────────┤
   ├─envsav───────────────────────────────┤
   ├─envset───────────────────────────────┤
   ├─f── ─data_set────────────────────────┤
   ├─P────────────────────────────────────┤
   │ ┌─53─────────────┐                   │
   ├─┴─p── ─port──────┴───────────────────┤
   │ ┌─nostick─┐                          │
   ├─┴─stick───┴──────────────────────────┤
   │ ┌─0──────────────────┐               │
   ├─┴─T── ─seconds───────┴───────────────┤
   ├─t── ─query_type──────────────────────┤
   └─x── ─dotted_decimal_notation_address─┘
```

Parameters

@*server*
: Specifies the domain name or IP address of the name server to contact for the query. The default is the name server specified in the TCPIP.DATA data set. TSO DIG can use only IPv4 addresses.

If a domain name is specified, DIG uses the resolver library routines provided in the TCP/IP for MVS programming interface to map the name to an IP address.

domain_name
: Specifies the name of the domain for which information is requested. If the domain name does not exist in the default domain specified in the TCPIP.DATA data set, you must specify a fully qualified domain name.

qtype
: Specifies the type of query to be performed. DIG does not support the MAILA, MD, MF, and NULL query types. The wildcard query types are ANY, MAILB, and AXFR. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query types.

If the *qtype* option is omitted, the default query type is A (an address query).

qclass
: Specifies which network class to request in the query. DIG recognizes only the IN, CHAOS, HESIOD, and ANY network classes.

%*comment*
: Provides a means of including comments in a DIG command. Any characters following the percent (%) character up to the next space character (space or end-of-record) are ignored by DIG. This option is useful in batch data sets for annotating a command.

For example, using a dotted decimal notation IP address rather than a domain name removes any overhead associated with address mapping; however, this makes the command less readable. Therefore, in a batch data set you can include the domain name as a comment for readability.

+*queryoption*
: Interprets the string following the plus sign (+) character as a query option. Query options have the format:

parameter[=*value*]

and are a superset of the SET subcommand options for NSLOOKUP.

aaonly
Accepts only authoritative responses to queries.

noaaonly
Accepts all responses to queries. This option is the default.

addit
Prints the additional section of the response. The additional section contains resource records that have not been explicitly requested, but could be useful. See RFC 1035 for more information about this option. This option is the default.

noaddit
Does not print the additional section of the response.

answer
Prints the answer section of the response. The answer section contains the set of all resource records from the name server database that satisfy the query. This option is the default.

noanswer
Does not print the answer section of the response.

author
Prints the authoritative section of the response. The authoritative section contains resource records that specify the address of an authoritative name server for the query. This section is used when the name server queried cannot provide an authoritative answer. This option is the default.

noauthor
Does not print the authoritative section of the response.

cl
Prints network class information for each of the resource records returned.

nocl
Does not print network class information for each of the resource records returned. This option is the default.

cmd
Echos the parsed options. This option is the default.

nocmd
Does not echo the parsed options.

d2
Prints the details of each query sent out to the network, including send time stamp and the timeout time stamp. When a server does not respond within the timeout period, DIG either sends the query to another server, or resends the query to the original server. The details of the query are visible when *d2* is set.

Note: You will not see any difference between debug, d2 and trace resolver. Resolver DNS responses and queries are traced for both options.

nod2
Does not print the details of each query sent out to the network. This option is the default.

debug
Directs DIG to print additional error messages. This option is the default.

nodebug
Directs DIG to not print additional error messages.

defname
Appends the default domain name to all unqualified domain names in a query. The default domain name is set by specifying the *+domain=name* option. This option is the default.

nodefname
Does not append the default domain name to all unqualified domain names in a query. This option causes the domain name specified to pass to the server without modification.

domain=*name*
Sets the default domain name to *name*. Initially the default domain name is obtained from the TCPIP.DATA data set. The validity of *name* is not verified. If the *defname* option is set, the domain name specified in *name* is appended to all unqualified domain names before the queries are sent to the name server.

Header
Prints the header line containing the operation code, returned status, and query identifier of each response. This option is distinct from the *header* option. This option is the default.

noHeader
Does not print the header line containing the operation code, returned status, and query identifier of each response.

header
Prints the query flags of each response. The query flags are defined in RFC 1035. This option is the default.

noheader
Does not print the query flags of each response.

ignore
Ignores truncation errors. Truncation errors occur when a response is too long for a single datagram.

noignore
Reports truncation errors. This option is the default.

ko
Keeps the virtual circuit open for queries in batch mode only. This option has no effect when used on the command line or when datagrams are used to transport queries (see the *novc* option later in this material).

noko
Does not keep the virtual circuit open for queries in batch mode only. This option is the default.

pfand=*number*
Performs a bitwise AND of the current print flags with the value specified in *number*. The number can be octal, decimal, or hexadecimal.

Note: To specify a number in octal, a 0 is required in front of the number. To specify a number in hexadecimal, 0X is required in front of the number.

pfdef
Sets the print flags to their default values. The default print flag values are 0x2FF9. For query type AXFR, the print flag values are 0x24F9.

Note: To specify a number in octal, a 0 is required in front of the number. To specify a number in hexadecimal, 0X is required in front of the number.

pfmin

Sets the print flags to the minimum default values. This option specifies that minimal information is printed for each response. The minimum print flag values are `0xA930`.

Note: To specify a number in octal, a 0 is required in front of the number. To specify a number in hexadecimal, 0X is required in front of the number.

pfor=*number*

Performs a bitwise OR of the current print flags with the value specified in *number*. The number can be octal, decimal, or hexadecimal.

Note: To specify a number in octal, a 0 is required in front of the number. To specify a number in hexadecimal, 0X is required in front of the number.

pfset=*number*

Sets the print flags to the value specified in *number*. The number can be octal, decimal, or hexadecimal.

Note: To specify a number in octal, a 0 is required in front of the number. To specify a number in hexadecimal, 0X is required in front of the number.

The print flags are represented by a 16-bit value. The following list describes the individual bits of the print flags in order of most-significant bit to least-significant bit.

0	Sort reply records
1	Unused
2	Display reply section
3	Display query section
4	Show basic header
5	Display time to live (TTL) in reply records
6	Show flags for query and reply
7	Show section headers with reply record totals
8	Show additional subsections
9	Show authoritative subsection
10	Show answer subsections
11	Show question subsections
12	Echo DIG command line
13	Display query class info in reply records
14	Unused
15	Display statistics

primary

Includes only the primary name server for the zone, or includes the secondary name servers.

noprimary
Indicates that you should not use only the primary name server for the zone. This option is the default.

qr
Prints the outgoing query. The outgoing query consists of the header and the question, empty answer, additional, and authoritative sections. See RFC 1035 for more information about outgoing queries.

noqr
Does not print the outgoing query. This option is the default.

ques
Prints the question section of a response. The question section contains the original query. This option is the default.

noques
Does not print the question section of a response.

recurse
Requests a recursive query when querying a name server. This option is the default.

norecurse
Specifies that a recursive query is not requested.

reply
Prints the response from the name server. This option is the default.

noreply
Does not print the response from the name server. When this option is disabled, other print flags that affect printing of the name server response are ignored and no sections of the response are printed.

retry=*limit*
Specifies the number of times a request is resent. When a request is sent and the timeout period expires for a response, the request is resent until the value specified in *limit* has been exceeded. The value specified in *limit* determines the number of attempts made to contact the name server. The default value for *limit* is retrieved from the TCPIP.DATA data set.

Setting *limit* to 0 disables DIG from contacting the name server. The result is an error message `no response from server`.

The retry procedure for DIG uses both the *limit* value and the timeout period. Each time a request is resent, the timeout period for the request is twice the timeout period used for the last attempt.

sort
Sorts resource records before printing. Records are sorted alphabetically on record type names.

nosort
Does not sort resource records before printing. This option is the default.

stats
Prints the query statistics including time and date of query, size of query and response packets, and name of server used. This option is the default.

nostats
Does not print the query statistics.

timeout=*time_out_value*
Specifies the number of seconds to wait before timing out of a request. The default timeout value is retrieved from the data set.

ttlid
Prints the time to live (TTL) for each resource record in a response. This option is the default.

nottlid
Does not print the TTL for each resource record in a response.

vc Uses a virtual circuit (TCP connection) to transport queries to the name server or datagrams. The default is retrieved from the TCPIP.DATA data set.

novc
Does not use a virtual circuit to transport queries to the name server or datagrams. This option is the default.

-*digoption*
Interprets the string following the hyphen (-) as a DIG option. The DIG options are either a parameter or a single character followed by a parameter.

c *query_class*
Specifies that the command-mode query or batch query retrieves resource records having the given network class. The *qclass* parameter, described in this topic, can also be used to specify the query class. In addition to the mnemonics, this option also accepts the equivalent numeric value that defines the class.

envsav
Directs DIG to save the environment specified on the current command line in the *user_id*.DIG.ENV data set. The DIG environment is described in "DIG internal state information" on page 921. This *hlq*.DIG.ENV data set initializes the default environment each time DIG is invoked.

envset
This option is valid for batch mode only. It directs DIG to set the default environment (see "DIG internal state information" on page 921) specified on the current line in the batch data set. This default environment remains in effect for all subsequent queries in the batch data set, or until the next line in the batch data set containing the *-envset* option is reached.

f *data_set*
Specifies a data set for DIG batch mode queries. The batch data set contains a list of queries that are to be executed in order. The keyword DIG is not used when specifying queries in a batch data set. Lines beginning with a number character (#) or semicolon (;) in the first column are comment lines, and blank lines are ignored. Options that are specified on the original command line are in effect for all queries in the batch data set unless explicitly overwritten. The following example shows a batch data set.

```
# A comment
; more comments
wurrup any in +noH =noqu -c IN

toolah +pfmin
```

Note: You must limit your query string to 99 characters to avoid error messages.

P Directs DIG to execute a PING command for response time comparison after receiving a query response. The last three lines of output from the following command are printed after the query returns:

```
PING server_name ( Length 56 Count 3
```

p *port*
Use the port number given when contacting the name server. The Domain Name System is a TCP/IP well-known service and has been allocated port 53. DIG uses port 53 by default, but this option enables you to override the port assignment.

stick
Restores the default environment (see "DIG internal state information" on page 921) before processing each line of a batch data set. This flag is valid for batch mode only. If you set the *stick* option, queries in the batch data set are not affected by the options specified for preceding queries in the data set.

nostick
Causes the query option specified on the current line in the batch data set to remain in effect until the option is overridden by a subsequent query. The result of each query in the batch data set depends on the preceding queries. This option is the default.

T *seconds*
Specifies the wait time between successive queries when operating in batch mode. The default wait time is 0 (do not wait).

t *query_type*
Specifies that the query retrieves resource records having the given resource record type. The *qtype* parameter, described in this topic, can also be used to specify the query type. In addition to the mnemonics, this parameter also accepts the equivalent numeric value that defines the type.

x *dotted_decimal_notation_address*
Simplifies the specification of a query for the `in-addr.arpa` domain. Normally these queries are made by specifying a query type of `PTR` for `nn.nn.nn.nn.in-addr.arpa`, where the four `nn` components are replaced by the dotted decimal notation IP address components in reverse order. This option enables you to make this query by simply specifying the dotted decimal notation IP address.

For example, the domain name corresponding to IP address `101.3.100.2` is found by a query for the domain name `2.100.3.101.in-addr.arpa`. You can use `DIG -x 101.3.100.2` rather than reversing the address and appending `in-addr.arpa`.

Examples

The following examples show how to use DIG to extract information from a name server. In Figure 3 on page 931, the router `wurrup` has two IP addresses, and there are two name servers, `wurrup` being the primary name server. This network is described by a single zone in the domain naming hierarchy stored in the name servers.

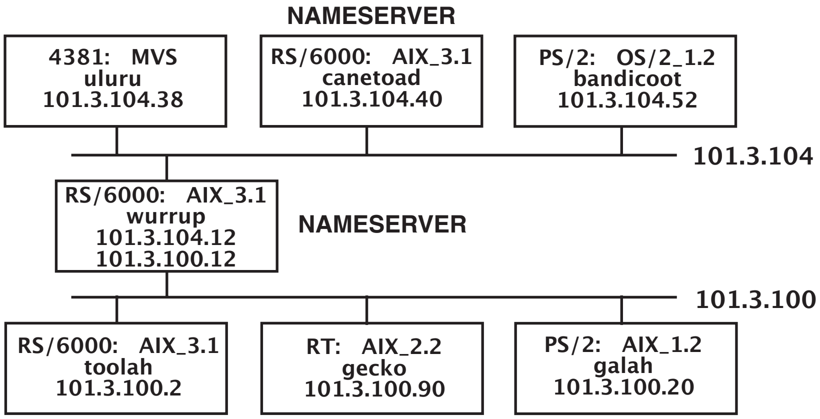


Figure 3. A TCP/IP network

In the examples, all queries are issued from the MVS `uluru` system.

Create a default environment (default options) that gives minimal output from subsequent DIG commands:

```
System:
Ready
User:
DIG wurrup +noqu +noH +nohe +nocmd +noad +noau +nost +nocl
+nottl -envsav
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

The following queries show which part of the response output is controlled by each of the output control options. Each example enables or disables query options for tailoring output.

- Set the query type to ns, the query class to in, and print the additional section of the output:

```
System:   Ready
User:
DIG fourex.oz ns in +ad
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 3
          ;; ANSWERS:
          fourex.oz NS      wurrup.fourex.oz
          fourex.oz NS      canetoad.fourex.oz
          ;; ADDITIONAL RECORDS:
          wurrup.fourex.oz  A       101.3.100.12
          wurrup.fourex.oz  A       101.3.104.12
          canetoad.fourex.oz        A       101.3.104.40
```

- Set the query type to ns, the query class to in, print the additional section of the output, but do not print the answer section:

```
System:   Ready
User:
DIG fourex.oz ns in +addit +noanswer
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 3
          ;; ADDITIONAL RECORDS:
          wurrup.fourex.oz  A       101.3.100.12
          wurrup.fourex.oz  A       101.3.104.12
          canetoad.fourex.oz        A       101.3.104.40
```

- Query a nonexistent domain and print the authoritative section of the output:

```
System:   Ready
User:
DIG noname +author
System:   ;; ->>HEADER<<- opcode: QUERY , status: NXDOMAIN, id: 3
          ; Ques: 1, Ans: 0, Auth: 1, Addit: 0
          ;; AUTHORITY RECORDS:
          fourex.oz SOA     wurrup.fourex.oz  adb.wurrup.fourex.oz (
                                 10003   ;serial
                                 3600    ;refresh
                                 300     ;retry
                                 3600000 ;expire
                                 86400 ) ;minim
```

In the previous example, the nonexistent domain name is *noname*.

- Use the default query options:

```
System:   Ready
User:
DIG wurrup
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Print the network class information:

```
System:   Ready
User:
DIG wurrup +cl
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  IN   A   101.3.104.12
          wurrup.FOUREX.OZ.  IN   A   101.3.100.12
```

- Echo the input query:

```
System:   Ready
User:
DIG wurrup +cmd
System:   ; <<>> DIG 2.0 <<>> wurrup +cmd
          ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Print the question section of the output:

```
System:   Ready
User:
DIG wurrup +qu
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup.FOUREX.OZ, type = A, class = IN

          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Turn the header on:

```
System:   Ready
User:
DIG wurrup +H
System:   ;;>>HEADER<<- opcde: QUERY , status: NOERROR, id: 3
          ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Print the query flags:

```
System:   Ready
User:
DIG wurrup +he
System:   ;; flags: qr aa rd ra ; Ques: 1, Ans: 2, Auth: 0, Addit:
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Print the question section and the outgoing query:

```
System:   Ready
User:
DIG wurrup +qu +qr
System:   ; Ques: 1, Ans: 0, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup.FOUREX.OZ, type = A, class = IN

          ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup.FOUREX.OZ, type = A, class = IN

          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Print the query statistics:

```
System:   Ready
User:
DIG fourex.oz ns in +stats
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 3
          ;; ANSWERS:
          fourex.oz NS      wurrup.fourex.oz
          fourex.oz NS      canetoad.fourex.oz
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:06:40 1992
          ;; MSG SIZE  sent: 24  rcvd: 116
```

- Print the TTL for each resource record:

```
System:   Ready
User:
DIG fourex.oz ns in +ttlid
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 3
          ;; ANSWERS:
          fourex.oz 9999999 NS      wurrup.fourex.oz
          fourex.oz 9999999 NS      canetoad.fourex.oz
```

- Enable extra debugging mode:

```
System:   Ready
User:
DIG wurrup +d2
System:   ;; res_mkquery(0, wurrup, 1, 1)
          ;; Querying server (# 1) address = 101.3.104.40
          ;; id = 3 - sending now: 4044656426 msec
          ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

The following examples show how options control the use and value of the default domain.

- Do not append the default domain name to unqualified domain names and print the question section of the response:

```
System:   Ready
User:
DIG wurrup +nodefname +qu
System:   ;;>>HEADER<<- opcde: QUERY , status: SERVFAIL, id: 3
          ; Ques: 1, Ans: 0, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup, type = A, class = IN
```

- Set the default domain name to fourexpd and print the question section of the response:

```
System:   Ready
User:
DIG wurrup +do=fourexpd +qu
System:   ;; ->>HEADER<<- opcode: QUERY , status: SERVFAIL, id: 3
          ; Ques: 1, Ans: 0, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup.fourexpd, type = A, class = IN
```

- Set the query type to ns, the query class to in and sort the output:

```
System:   Ready
User:
DIG fourex.oz ns in +sort
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 3
          ;; ANSWERS:
          fourex.oz NS      canetoad.fourex.oz
          fourex.oz NS      wurrup.fourex.oz
```

- Query the domain at the address 101.3.100.20, and print the question section of the response:

```
System:   Ready
User:
DIG -x 101.3.100.20 +qu
System:   ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;    20.100.3.101.in-addr.arpa, type = ANY, class = IN

          ;; ANSWERS:
          20.100.3.101.in-addr.arpa.      PTR     galah.
```

- Retrieve resource records with a network class of ANY and print the question section of the response:

```
System:   Ready
User:
DIG wurrup -c any +qu
System:   ; Ques: 1, Ans: 2, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup.FOUREX.OZ, type = A, class = ANY

          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
```

- Retrieve resource records with a query type of ANY and print the question section of the response:

```
System:   Ready
User:
DIG wurrup -t any +qu
System:   ; Ques: 1, Ans: 3, Auth: 0, Addit: 0
          ;; QUESTIONS:
          ;;      wurrup.FOUREX.OZ, type = ANY, class = IN

          ;; ANSWERS:
          wurrup.FOUREX.OZ.  A       101.3.104.12
          wurrup.FOUREX.OZ.  A       101.3.100.12
          wurrup.FOUREX.OZ.  HINFO   RS6000 AIX3.1
```

The following lists the batch data set, `test.digbat`, used for this example. The default environment has been removed by discarding the *user_id*.DIG.ENV data set. The DIG command is omitted for all entries in the data set.

Note the effect of the *-envset* and *-stick* options on the output:

```
wurrup any in +noH +nohe +noqu +noad +noau -envset -stick
wurrup any in
toolah a in +d2
toolah a in
toolah a in +d2 -nostick
toolah a in
toolah a in +nod2
toolah a in
```

Specify the batch data set test.digbat:

```
System:  Ready
User:
DIG -f test.digbat

System:   ; <<>> DIG 2.0 <<>> DIG wurrup any in +noH +nohe +noqu +noad
          +noau -envset -stick
          ; Ques: 1, Ans: 3, Auth: 0, Addit: 0

          ;; ANSWERS:
          wurrup.FOUREX.OZ.  9999999 A       101.3.104.12
          wurrup.FOUREX.OZ.  9999999 A       101.3.100.12
          wurrup.FOUREX.OZ.  86400   HINFO   RS6000 AIX3.1

          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 95
```

```
System:   ; <<>> DIG 2.0 <<>> DIG wurrup any in
          ; Ques: 1, Ans: 3, Auth: 0, Addit: 0
          ;; ANSWERS:
          wurrup.FOUREX.OZ.  9999999 A       101.3.104.12
          wurrup.FOUREX.OZ.  9999999 A       101.3.100.12
          wurrup.FOUREX.OZ.  86400   HINFO   RS6000 AIX3.1
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 95
```

```
System:   ; <<>> DIG 2.0 <<>> DIG toolah a in +d2
          ;; res_mkquery(0, toolah, 1, 1)
          ;; Querying server (# 1) address = 101.3.104.40
          ;; id = 3 - sending now: 4046124888 msec
          ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; ANSWERS:
          toolah.FOUREX.OZ.  9999999 A       101.3.100.2
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 47
```

```
System:   ; <<>> DIG 2.0 <<>> DIG toolah a in
          ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; ANSWERS:
          toolah.FOUREX.OZ.  9999999 A       101.3.100.2
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 47
```

```
System:   ; <<>> DIG 2.0 <<>> DIG toolah a in +d2 -nostick
          ;; res_mkquery(0, toolah, 1, 1)
          ;; Querying server (# 1) address = 101.3.104.40
          ;; id = 3 - sending now: 4046125037 msec
          ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; ANSWERS:
          toolah.FOUREX.OZ.  9999999 A    101.3.100.2
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 47
```

```
System:   ; <<>> DIG 2.0 <<>> DIG toolah a in
          ;; res_mkquery(0, toolah, 1, 1)
          ;; Querying server (# 1) address = 101.3.104.40
          ;; id = 5 - sending now: 4046125101 msec
          ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; ANSWERS:
          toolah.FOUREX.OZ.  9999999 A       101.3.100.2
           ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 47
```

```
System:   ; <<>> DIG 2.0 <<>> DIG toolah a in +nod2
          ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; ANSWERS:
          toolah.FOUREX.OZ.  9999999 A       101.3.100.2
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:57 1992
          ;; MSG SIZE  sent: 31  rcvd: 47
```

```
System:   ; <<>> DIG 2.0 <<>> DIG toolah a in
          ; Ques: 1, Ans: 1, Auth: 0, Addit: 0
          ;; ANSWERS:
          toolah.FOUREX.OZ.  9999999 A       101.3.100.2
          ;; FROM: FOUREXVM1 to SERVER: default -- 101.3.104.40
          ;; WHEN: Tue Mar 16 11:15:58 1992
          ;; MSG SIZE  sent: 31  rcvd: 47
```

Usage

The *queryoption* and *digoption* parameters are case sensitive and must be entered in lowercase. Domain names, query types, query classes, and the values associated with *queryoption* and *digoption* parameters are not case sensitive.

Using the z/OS UNIX dig command

The domain information groper (**dig**) is a command line tool that can be used to gather information from the Domain Name System servers. The **dig** command has two modes: simple interactive mode for a single query, and batch mode, which executes one query for each in a list of several query lines. All query options are accessible from the command line.

Rule: The dig command does not use resolver caching.

The **dig** command is used to query Domain Name Servers, which enables you to:

- Exercise name servers
- Gather large volumes of domain name information
- Execute simple domain name queries
- Execute multiple lookups from the command line

dig command: Query name servers

You can use **dig** in several methods:

- **Command Line**

  All options are specified on the invoking command line.
- **Batch Mode**

A group of queries are placed in a file and executed by a single invocation of **dig** using the **-f** *filename* option. The *filename* contains complete queries, one per line. The keyword *dig* is not used within a batch file when specifying queries. Blank lines are ignored, and lines beginning with a # character or a semicolon (;) in the first column are comment lines.

- **Multiple Queries**

  The BIND 9 implementation of **dig** supports specifying multiple queries on the command line (in addition to supporting the **-f** batch file option). Each of those queries can be supplied with its own set of flags, options and query options. In multiple queries, *query1*, *query2*, and so on represent an individual query in the command-line syntax. Each consists of any of the standard options and flags, the name to be looked up, an optional query type and class and any query options that are applied to that query.

  Note: When entered on a z/OS UNIX shell command line, long **dig** commands can be broken into segments entered with a terminating backslash (\) except for the last segment.

  A global set of query options, which are applied to all queries, can also be supplied. These global query options must precede the first query set (name, class, type, options, flags, and query options) supplied on the command line. Any global query options can be overridden by a query-specific set of query options.

Options specified on the initial command line are in effect for all queries in the batch file unless explicitly overridden. Several options are provided exclusively for use within batch data sets, giving greater control over **dig** operation.

If a name server is not specified, **dig** tries each of the servers found in its TCPIP.DATA statements. When no command line arguments or options are given, **dig** performs an NS query for "." (the root).

Some of **dig** initial settings are retrieved from TCPIP.DATA, according to the resolver search order. See the z/OS Communications Server: IP Configuration Guide for more information about the search order for finding TCPIP.DATA statements. It uses directives from the resolver configuration file in the following order:

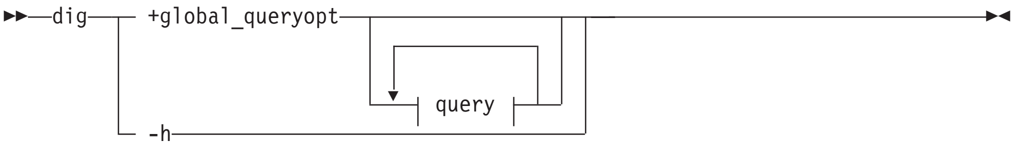
1. nameserver/nsinteraddr
2. options ndots:n
3. search
4. domain/domainorigin

Format

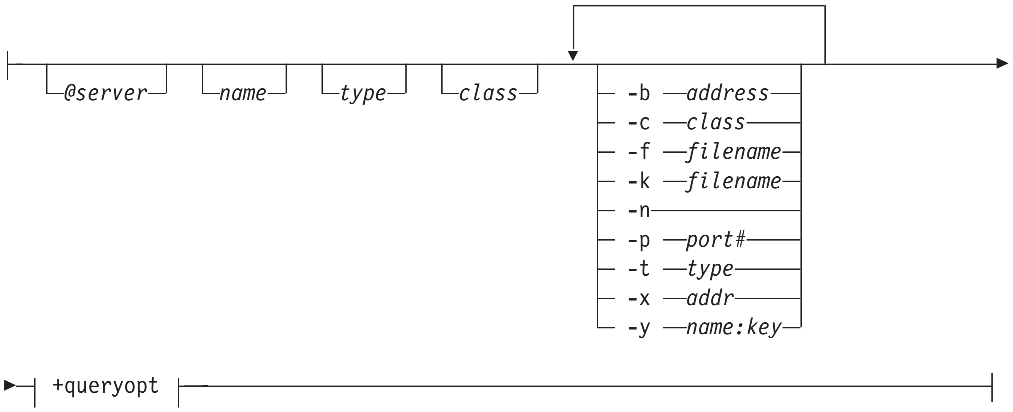
Command Line Mode

```
►►──dig──┬──────────────┬──────────────────────────────────────────►◄
         ├─┤ query ├────┤
         └─-h───────────┘
```

Multiple Query Mode



query:



+queryopt or +global_queryopt:

```
     ┌──────────────────────────────────┐
     ▼                                  │
├────┬─ +noaaonly │ +aaonly──────────────┬─┴──────────────────────────────────────┤
     ├─ +noadditional │ +additional──────┤
     ├─ +noadflag │ +adflag──────────────┤
     ├─ +noall │ +all────────────────────┤
     ├─ +noanswer │ +answer──────────────┤
     ├─ +noauthority │ +authority────────┤
     ├─ +nobesteffort │ +besteffort──────┤
     ├─ +nocdflag │ +cdflag──────────────┤
     ├─ +nocmd │ +cmd────────────────────┤
     ├─ +nocomments │ +comments──────────┤
     ├─ +nodefname │ +defname────────────┤
     ├─ +nodnssec │ +dnssec──────────────┤
     ├─ +nofail │ +fail──────────────────┤
     ├─ +noidentify │ +identify──────────┤
     ├─ +noignore │ +ignore──────────────┤
     ├─ +nomultiline │ +multiline────────┤
     ├─ +nonssearch │ +nssearch──────────┤
     ├─ +noqr │ +qr──────────────────────┤
     ├─ +noquestion │ +question──────────┤
     ├─ +norecursive │ +recursive────────┤
     ├─ +nosearch │ +search──────────────┤
     ├─ +noshort │ +short────────────────┤
     ├─ +nosta │ +sta────────────────────┤
     ├─ +notcp │ +tcp────────────────────┤
     ├─ +notrace │ +trace────────────────┤
     ├─ +novc │ +vc──────────────────────┤
     ├─ +bufsize──=──B───────────────────┤
     ├─ +domain──=──somename─────────────┤
     ├─ +ndots──=──D─────────────────────┤
     ├─ +time──=──T──────────────────────┤
     └─ +tries──=──A─────────────────────┘
```

Parameters

-h Provides help for the **dig** command.

@server

The name or IP address of the name server to query. An IPv4 or IPv6 address or a name that resolves to an IPv4 or IPv6 address can be specified. When the supplied server argument is a host name, **dig** resolves that name before querying that name server. If no server argument is provided, **dig** consults TCPIP.DATA statements and queries the name servers listed there. The reply from the name server that responds is displayed.

name

The name of the resource record that is to be looked up.

type

Specifies what type of query is required. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query types.

If the *type* option is omitted, the default query type is A (an address query).

class

Specifies which network class to request in the query. **dig** recognizes only the IN, CHAOS, HESIOD, and ANY network classes. The default class is IN. See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query classes.

-query_options
These options must be preceded by a minus (**-**) sign.

-b *address*
Sets the source IP address of the query to *address*. This must be a valid address on one of the host's network interfaces. An IPv6 address can be used here only if the address of the name server is also an IPv6 address. In order to accomplish this, the IPv6 name server address must be explicitly specified with the @ symbol.

-c *class*
Overrides the default query class (IN for Internet). See the z/OS Communications Server: IP Configuration Reference for detailed information about valid query classes.

-f *filename*
Makes **dig** operate in batch mode by reading a list of lookup requests to process from the file *filename*. The file contains a number of queries, one per line. Organize each entry in the file in the same way they would be presented as queries to **dig** using the command line interface.

-k *filename*
Specifies a TSIG key *filename* to sign the DNS queries sent by **dig** and their responses using transaction signatures (TSIG).

-n Sends the query for the IPv6 address specified on the **-x** option as a *nibble* label in the `IP6.ARPA` domain.

-p *port#*
This option would be used to test a name server that has been configured to listen for queries on a non-standard port number. **dig** will send its queries to *port#*. The standard DNS port number is 53.

-s Sends the reverse query for the IPv6 address specified on the **-x** option as a bitstring label in the `IP6.ARPA` domain.

-t *type*
Sets the query type to *type*. It can be any valid query type supported in BIND 9. The default query type is A, unless the **-x** option is supplied to indicate a reverse lookup. A zone transfer can be requested by specifying a type of AXFR. When an incremental zone transfer (IXFR) is required, type is set to ixfr=N. The incremental zone transfer will contain the changes made to the zone because the serial number in the zone's SOA record was N.

-x *addr*
Reverses lookups by mapping addresses to names. *addr* is an IPv4 address in dotted decimal notation, or an IPv6 address in colon hexadecimal notation. When this option is used, there is no need to provide the name, class and type arguments. **dig** automatically performs a lookup for a name like `11.12.13.10.in-addr.arpa` and sets the query type and class to PTR and IN respectively. By default, IPv6 addresses are looked up using the `IP6.ARPA` domain and binary labels as defined in RFC 2874. To use the older RFC 1886 method using the `IP6.ARPA` domain and *nibble* labels, specify the **-n** (nibble) option.

-y *name:key*
You can use this option to specify the TSIG key itself on the command line. *name* is the name of the TSIG key and *key* is the actual key. The key is a base-64 encoded string, typically generated by dnssec-keygen. Take care when using this option on multiuser systems as the key can be visible in

the output from `ps -ef` or in the shell's history file. When using TSIG authentication with **dig**, the name server that is queried needs to know the key and algorithm that is being used. In BIND 9, this is done by providing appropriate key{} and server{} statements in *named.conf*.

+queryopt

The query options available in the **dig** command. These options must be preceded by a plus (**+**) sign. Many of these options can be abbreviated by the minimum unique prefix string that is usually two characters, but three for **+additional** and **+adflag**. To abbreviate the negative command, prepend the unique string with **no**. Some of these set or reset flag bits in the query header, some determine which sections of the answer get printed, and others determine the timeout and retry strategies.

When used in multiple queries, **+queryopt** options can become a global options (**+queyoption_global**). To be a valid global option, **+queyoption_global** must be placed before the first query set to be queried.

+[no]aaonly

This option does nothing. It is provided for compatibility with old versions of **dig** where it set an unimplemented resolver flag.

+[no]additional

Display [do not display] the additional section of a reply. The default is to display it.

+[no]adflag

Set [do not set] the AD (authentic data) bit in the query. The AD bit currently has a standard meaning only in responses, not in queries, but the ability to set the bit in the query is provided for completeness.

+[no]all

Set or clear all display flags. The default is on.

+[no]answer

Display [do not display] the answer section of a reply. The default is to display it.

+[no]authority

Display [do not display] the authority section of a reply. The default is to display it.

+[no]besteffort

Try [do not try] to parse illegal messages. The default is not to parse illegal messages.

+[no]cdflag

Set [do not set] the CD (checking disabled) bit in the query. This requests the server to not perform DNSSEC validation of responses. The default is off, meaning that DNSSEC validation will occur.

+[no]cmd

Toggles the printing of the initial comment in the output identifying the version of **dig** and the query options that have been applied. This comment is printed by default. This option is recognized only when used as a global option (placed before the first query).

+[no]comments

Toggle the display of comment lines in the output. The default is to print comments.

+[no]defname
Use [do not use] the default domain name, if any, in TCPIP.DATA. The default is not to append that name to name when making queries.

+[no]dnssec
Request [do not request] DNSSEC records. The default is not to request DNSSEC records.

+[no]fail
Try the next server on SERVFAIL (fail), or do not try the next server on SERVFAIL (nofail). The default is not to try the next server on SERVFAIL.

+[no]identify
Show [do not show] the IP address and port number that supplied the answer when the +short option is enabled. If short form answers are requested, the default is not to show the source address and port number of the server that provided the answer.

+ignore
Ignore truncation in UDP responses instead of trying again with TCP. By default, TCP retries are performed.

+[no]multiline
Print [do not print] records in expanded format. The default is not to print records in expanded format.

+[no]nssearch
When this option is set on, **dig** attempts to find the authoritative name servers for the zone containing the name being looked up and display the SOA record that each name server has for the zone. The default is off.

+[no]qr
Print [do not print] the query as it is sent before sending the query. By default, the query is not printed.

+[no]question
Print [do not print] the question section of a query when an answer is returned. The default is to print the question section as a comment.

+[no]recursive
Toggle the setting of the RD (recursion required) bit in the query. This bit is set by default, which means **dig** normally sends recursive queries. Recursion is automatically disabled when the +nssearch or +trace query options are used.

+[no]search
Use [do not use] the search list in TCPIP.DATA. The search list is not used by default.

+[no]short
Provide a terse answer. The default is to print the answer in a verbose form.

+[no]sta
This query option toggles the printing of statistics when the query was made, the size of the reply and so on. The default behavior is to print the query statistics.

+[no]tcp
Use [do not use] TCP when querying name servers. The default is UDP unless an AXFR or IXFR query is requested, in which case a TCP connection is used.

+[no]trace
Toggle tracing of the delegation path from the root name servers for the name being looked up. Tracing is disabled by default. When tracing is enabled, **dig** makes iterative queries to resolve the name being looked up. It will follow referrals from the root servers, showing the answer from each server that was used to resolve the lookup.

+[no]vc
Use [do not use] TCP virtual circuit when querying name servers. This alternate syntax to +[no]tcp is provided for backwards compatibility. By default, UDP is used instead of TCP.

+bufsize=B
Set the UDP message buffer size advertised using EDNS0 to *B* bytes. The maximum and minimum sizes of this buffer are 65 535 and 0 respectively. Values outside this range are rounded up or down appropriately. The default value is 2048.

+domain=*somename*
Set the default domain to *somename*, as if specified in a domain directive or domainorigin in the resolver configuration file.

+ndots=*D*
Set the number of dots that have to appear in name to be considered absolute. The default value is that defined using the ndots statement in resolver configuration file, or 1 if ndots statement is not present. Names with fewer dots are interpreted as relative names and are searched for in the domains listed in the search or domain/domainorigin directive in the resolver configuration file.

+time=*T*
Sets the timeout for a query to *T* seconds. The default timeout is 5 seconds. An attempt to set *T* to less than 1 will result in a query timeout of 1 second being applied.

+tries=*A*
Sets the number of times to retry UDP queries to server. The default number of tries is 3. If *T* is less than or equal to 0, the number of retries is set to 1.

Examples

The following examples show how to use **dig** to extract information from a name server.

Any global query options can be overridden by a query-specific set of query options.

```
dig +qr www.isc.org any -x 127.0.0.1 isc.org ns +noqr
```

Shows how **dig** could be used from the command line to make three lookups: an ANY query for `www.isc.org`, a reverse lookup of `127.0.0.1` and a query for the NS records of `isc.org`. A global query option of **+qr** is applied, so that **dig** shows the initial query it made for each lookup. The final query has a local query option of **+noqr** which means that **dig** will not print the initial query when it looks up the NS records for `isc.org`.

The following example shows a basic **dig** command, with default print options.

```
>dig @9.67.128.82 vic032.tcp.raleigh.ibm.com.
; <<>> DiG 9.2.0 <<>> @9.67.128.82 vic032.tcp.raleigh.ibm.com.
;; global options: printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49597
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; QUESTION SECTION:
 ;vic032.tcp.raleigh.ibm.com. IN A
 ;; ANSWER SECTION:
 vic032.tcp.raleigh.ibm.com. 86400 IN A 9.67.113.32
 ;; AUTHORITY SECTION:
 tcp.raleigh.ibm.com. 86400 IN NS buzz.tcp.raleigh.ibm.com.
 ;; ADDITIONAL SECTION:
 buzz.tcp.raleigh.ibm.com. 86400 IN A 9.67.128.82
 ;; Query time: 10 msec
 ;; SERVER: 9.67.128.82#53(9.67.128.82)
 ;; WHEN: Mon Apr 30 12:13:10 2001
 ;; MSG SIZE rcvd: 114
```

The following example shows a **dig** command with specified port, type, class, and short answer with identity of the response sender.

```
$>dig @9.67.113.32 version.bind -p 20321 ANY CH +short +identity
 Allocated socket 5, type udp
 ; <<>> DiG 9.2.0 <<>> @9.67.113.32 version.bind -p 20321 ANY CH +short
 +identity
 ;; global options: printcmd
 "9.2.0" from server 9.67.113.32 in 11 ms.
```

The following example shows a **dig** command with global options set for queries on two host names.

```
>dig @9.67.128.82 +noquestion +noauthority +noadditional +nosta
; <<>> DiG 9.2.0 <<>> @9.67.128.82 +noquestion +noauthority +noadditional +nosta
 +domain=tcp.raleigh.ibm.com vic032 mvs183
 ;; global options: printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49597
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; ANSWER SECTION:
 vic032.tcp.raleigh.ibm.com. 86400 IN A 9.67.113.32
 Allocated socket 6, type udp
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41218
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; ANSWER SECTION:
 mvs183.tcp.raleigh.ibm.com. 86400 IN A 9.37.65.154
```

The following example shows a v command where a set of global options apply only to the first query. The following two queries reverse some of the global option values (notice the + options follow the affected query name).

```
>dig @9.67.128.82 +noquestion +noauthority +noadditional +nosta
; <<>> DiG 9.2.0 <<>> @9.67.128.82 +noquestion +noauthority +noadditional +nosta
 +domain=tcp.raleigh.ibm.com vic032 mvs183 +question +authority mvs150 +additional
 +sta
 ;; global options: printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49597
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; ANSWER SECTION:
 vic032.tcp.raleigh.ibm.com. 86400 IN A 9.67.113.32
 Allocated socket 6, type udp
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41218
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; QUESTION SECTION:
```

```
 ;mvs183.tcp.raleigh.ibm.com. IN A
 ;; ANSWER SECTION:
 mvs183.tcp.raleigh.ibm.com. 86400 IN A 9.37.65.154
 ;; AUTHORITY SECTION:
 tcp.raleigh.ibm.com. 86400 IN NS buzz.tcp.raleigh.ibm.com.
 Allocated socket 5, type udp
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20635
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; ANSWER SECTION:
 mvs150.tcp.raleigh.ibm.com. 3600 IN A 9.67.113.117
 ;; ADDITIONAL SECTION:
 buzz.tcp.raleigh.ibm.com. 86400 IN A 9.67.128.82
 ;; Query time: 16 msec
 ;; SERVER: 9.67.128.82#53(9.67.128.82)
 ;; WHEN: Mon Apr 30 15:27:26 2001
 ;; MSG SIZE rcvd: 114
```

The following example shows a **dig** command for 2 queries where type and class default values are overridden with new values for the second query.

```
>dig @9.67.128.82 +noquestion +noauthority +noadditional +nosta
; <<>> DiG 9.2.0 <<>> @9.67.128.82 +noquestion +noauthority +noadditional +nosta
 +domain=tcp.raleigh.ibm.com vic032 version.bind -t txt -c ch +question
 ;; global options: printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49597
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
 ;; ANSWER SECTION:
 vic032.tcp.raleigh.ibm.com. 86400 IN A 9.67.113.32
 Allocated socket 6, type udp
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41218
 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
 ;; QUESTION SECTION:
 ;version.bind. CH TXT
 ;; ANSWER SECTION:
 VERSION.BIND. 0 CH TXT "9.2.0"
```

The following example shows a **dig** command specifying an IPv6 address for the name server to query.

```
>dig @::1 ns .
; <<>> DiG 9.2.0 <<>> @::1 ns .
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32799
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 4

;; QUESTION SECTION:
;.    IN NS

;; ANSWER SECTION:
.   232344 IN NS J.ROOT-SERVERS.NET.
.   232344 IN NS K.ROOT-SERVERS.NET.
.   232344 IN NS L.ROOT-SERVERS.NET.
.   232344 IN NS M.ROOT-SERVERS.NET.
.   232344 IN NS A.ROOT-SERVERS.NET.
.   232344 IN NS B.ROOT-SERVERS.NET.
.   232344 IN NS C.ROOT-SERVERS.NET.
.   232344 IN NS D.ROOT-SERVERS.NET.
.   232344 IN NS E.ROOT-SERVERS.NET.
.   232344 IN NS F.ROOT-SERVERS.NET.
.   232344 IN NS G.ROOT-SERVERS.NET.
.   232344 IN NS H.ROOT-SERVERS.NET.
.   232344 IN NS I.ROOT-SERVERS.NET.
```

```
;; ADDITIONAL SECTION:
J.ROOT-SERVERS.NET. 369255 IN A 198.41.0.10
K.ROOT-SERVERS.NET. 369255 IN A 193.0.14.129
L.ROOT-SERVERS.NET. 318744 IN A 198.32.64.12
M.ROOT-SERVERS.NET. 318744 IN A 202.12.27.33

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Fri Jul 13 00:09:24 2001
;; MSG SIZE  rcvd: 292
```

Usage

The *queryoption* and *option* parameters are case sensitive and must be entered in lowercase. Domain names, query types, query classes, and the values associated with *queryoption* and *option* parameters are not case sensitive.

Using the z/OS UNIX host command

The z/OS UNIX host command queries the configured name server to perform the following tasks:

- Identify the IP addresses associated with a specified DNS host name
- Identify the DNS hostnames associated with a specified IP address

The host command must be issued from within the z/OS UNIX shell.

z/OS UNIX host: Identify the remote host

Use the z/OS UNIX host command to identify the IP addresses associated with a specified DNS host name or to identify the DNS hostnames associated with a specified IP address.

Format

```
►►—host—host——————————————————————————————————————————————————►◄
```

Parameters

host
DNS host name or IP address to look up

Examples

The following example shows the command output:

```
host 204.146.18.33
EZZ8321I www.ibm.com has addresses 204.146.18.33

host www.ibm.com
EZZ8321I www.ibm.com has addresses 204.146.18.33
```

Using the z/OS UNIX hostname command

Use the z/OS UNIX **hostname** command to display the fully-qualified DNS host name of the system. This command uses the following to determine this information:

- TCPIP.DATA statement information

- DNS lookup on the value that is returned by gethostname()
- The gethostname() function call, which is documented in z/OS XL C/C++ Runtime Library Reference.

TCPIP.DATA statement information

To display the TCPIP.DATA information found in the search order, you can use the z/OS Communications Server resolver trace support. To activate the trace, you can set the RESOLVER_TRACE environment variable before invoking the **hostname** command. Setting the environment variable to use stdout is recommended. The resolver trace is described in z/OS Communications Server: IP Diagnosis Guide.

See z/OS Communications Server: IP Configuration Reference for detailed information about TCPIP.DATA statements. The z/OS UNIX search order is always used to find the TCPIP.DATA statements for the **hostname** command. See information about search orders used in the z/OS UNIX environment in z/OS Communications Server: IP Configuration Guide for a description of this search order.

gethostname() function call

The gethostname() function call is processed by a TCP/IP stack. In a single-stack (INET) environment, the host name of the active stack is returned. In a multi-stack (CINET) environment, the gethostname() call is sent either to the stack with which the z/OS UNIX user has set affinity (for example, by setting the _BPXK_SETIBMOPT_TRANSPORT environment variable) or, if there is no stack affinity, it is sent to the default stack. The default stack in a CINET environment is determined by SUBFILESYSTYPE statements in the BPXPRMxx parmlib member. If the default stack is not active, the first stack that is activated is designated as the default stack. You can display the current default stack using the DISPLAY OMVS,PFS operator command. The FLAGS column on the command output indicates which stack is the default. See z/OS MVS System Commands for the DISPLAY command output. See z/OS UNIX System Services Planning for information about how z/OS selects the default stack and how stack affinity can be established.

The TCP/IP stack that receives the gethostname() function call returns the host name value that was determined during stack initialization. During initialization, the configuration component of the stack issues the __iphost() function call to get the TCPIP.DATA HOSTNAME statement value of the stack.

Note: If the HOSTNAME statement is changed, TCPIP needs to be restarted for this change to take effect.

The z/OS UNIX search order for the started task of the stack is used to find the TCPIP.DATA statement information of the stack. The host name is determined in the following order:

1. If the found TCPIP.DATA information contains a valid HOSTNAME statement, its value is returned. If a system name parameter was specified on the HOSTNAME statement, the parameter value is compared to one of the following to determine whether the value matches the current system name:
   - If you have configured VMCF and TNF as non-restartable subsystems, the system name is specified in the IEFSSN*xx* member of PARMLIB.
   - If you have configured VMCF and TNF as restartable subsystems, the system name is obtained from the value of the P= parameter of the EZAZSSI started procedure.

2. If there is no valid HOSTNAME statement, the VMCF node name with which VMCF was started is returned.
3. If VMCF was not active when the stack was started, the CVTSNAME value (this is the SYSNAME=value in the IEASYS*xx* member that was used during the IPL) is returned.

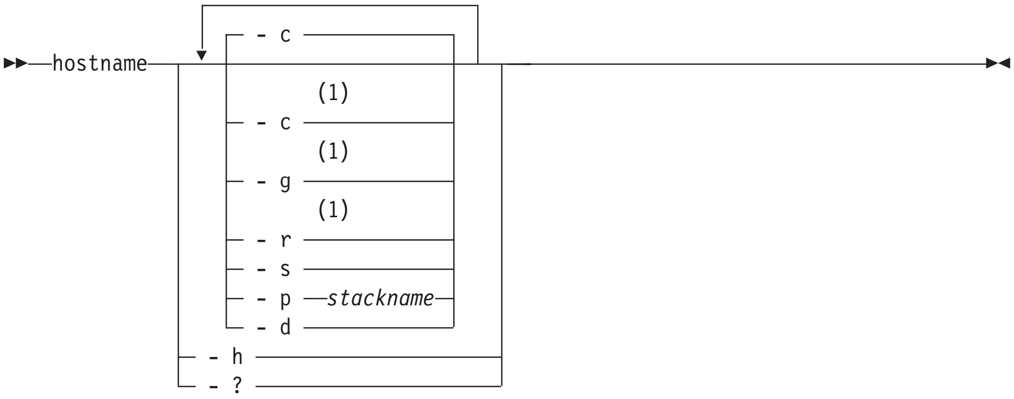
If the host name came from TCPIP.DATA, the case used is the case that was specified on the HOSTNAME statement. For VMCF or CVTSNAME, the name is in uppercase.

The z/OS UNIX **hostname** command must be issued from within the z/OS UNIX shell.

z/OS UNIX hostname: Identify the local host

Use the z/OS UNIX **hostname** command to display the fully qualified DNS host name of the local system.

Format



Notes:

1 Only one of the -c, -g, and -r parameters can be specified.

Parameters

-c Uses the TCPIP.DATA configuration (this is the default).

Specifies the fully qualified host name (the host name with its appended domain name). The host name is obtained with the __iphost() function call. The host name portion of the displayed name is the value of the z/OS UNIX user's TCPIP.DATA HOSTNAME statement. The domain name portion of the displayed name is the value of the z/OS UNIX user's TCPIP.DATA DomainOrigin statement or the first domain name that is specified by the SEARCH statement.

-g Uses gethostname() result.

Specifies the host name without the domain name. The host name is obtained by using the gethostname() function call. The displayed name is the value of the TCPIP.DATA HOSTNAME statement for the TCP/IP stack.

-r Uses DNS lookup on gethostname() result.

Specifies the fully qualified host name (the host name that is returned by a gethostname() function call), apppended with the domain name. The returned host name is the value of the TCPIP.DATA HOSTNAME statement for theTCP/IP stack. The gethostbyname() call uses the TCPIP.DATA resolver statements of the z/OS UNIX user to obtain the fully qualified host name. Based on those statements, Domain Name Servers (DNS) and resolver local host tables are used for the name resolution. The displayed name is the result of the name resolution.

-s Prints the short name of the host (without the DNS domain name).

Specifies the host name without the domain name. The host name is obtained by using the __iphost() function call. The displayed name is the value of the TCPIP.DATA HOSTNAME statement of the z/OS UNIX user.

-p *stackname*
Uses this AF_INET stack.

Specifies which TCP/IP stack the **-g** or **-r** parameter uses for its gethostname() function call.

-d Prints trace messages for problem diagnosis.

-h Displays the usage message.

-? Displays the usage message.

Using the z/OS UNIX dnsdomainname command

Use the z/OS UNIX **dnsdomainname** command to display the DNS domain name of the system. The **dnsdomainname** command uses the following for determining this information:

- TCPIP.DATA statement information
- DNS lookup on the value returned by gethostname()
- The gethostname() function call, which is documented in z/OS XL C/C++ Runtime Library Reference

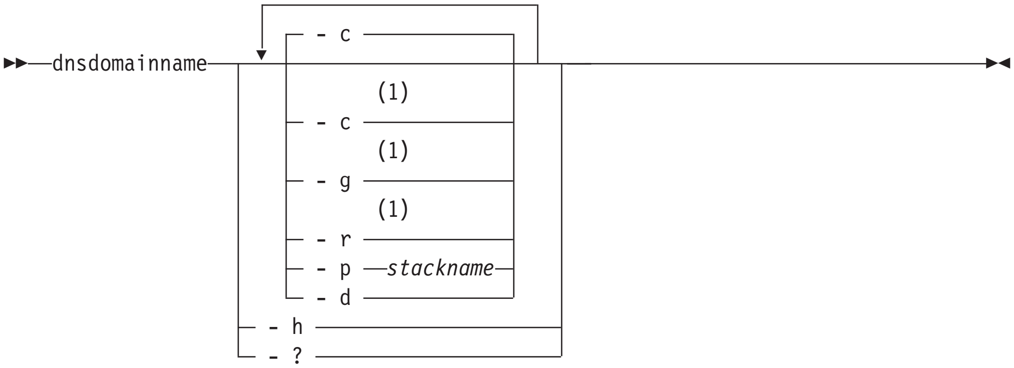
The **dnsdomainname** command and the **hostname** command are processed by the same function and support most of the same parameters. See “Using the z/OS UNIX hostname command” on page 947 for more detailed information about the methods used to provide the host and domain names. See the description of the **hostname** command parameters in “z/OS UNIX hostname: Identify the local host” on page 949 for a more detailed description of the **dnsdomainname** command parameters.

You must issue the z/OS UNIX **dnsdomainname** command from within the z/OS UNIX shell.

z/OS UNIX dnsdomainname: Display the DNS domain name

Use the z/OS UNIX **dnsdomainname** command to display the DNS domain name of the system.

Format



Notes:

1 Only one of the -c, -g, and -r parameters can be specified.

Parameters

-c uses the TCPIP.DATA configuration.

-g uses gethostname() result.

-r uses DNS lookup on gethostname() result.

-p *stackname*
uses this AF_INET stack.

-d prints trace messages for problem diagnosis.

-h displays the usage message.

-? displays the usage message.

Usage

- If the DNS domain name cannot be retrieved, an error message is displayed.

Using the z/OS UNIX domainname command

The z/OS UNIX **domainname** command is a synonym for the z/OS UNIX **dnsdomainname** command. See “Using the z/OS UNIX dnsdomainname command” on page 950 for information about using this command.

Note: On some operating systems, the **domainname** command displays the system NIS/YP domain name, which might or might not be the same as the system DNS domain name. Portable shell scripts must use **dnsdomainname** rather than **domainname** if this distinction is important.

Chapter 7. Managing TCP/IP network resources with SNMP

This information describes how to use the Simple Network Management Protocol (SNMP) commands and details what support the z/OS Communications Server SNMP agent and subagents provide.

The z/OS UNIX snmp command

The z/OS UNIX **snmp** command provides the following SNMP manager functions from the z/OS UNIX shell:

- Query SNMP agents for network management information.
- Receive and format SNMP traps and notifications.

Use the **snmp** command to issue SNMP requests to agents and to process SNMP responses returned by agents. This command supports a maximum SNMP response packet size of 65 535 bytes. SNMPv1, SNMPv2c, and SNMPv3 requests are supported.

Use the **snmp** command with the trap request type to receive and format traps and notifications.

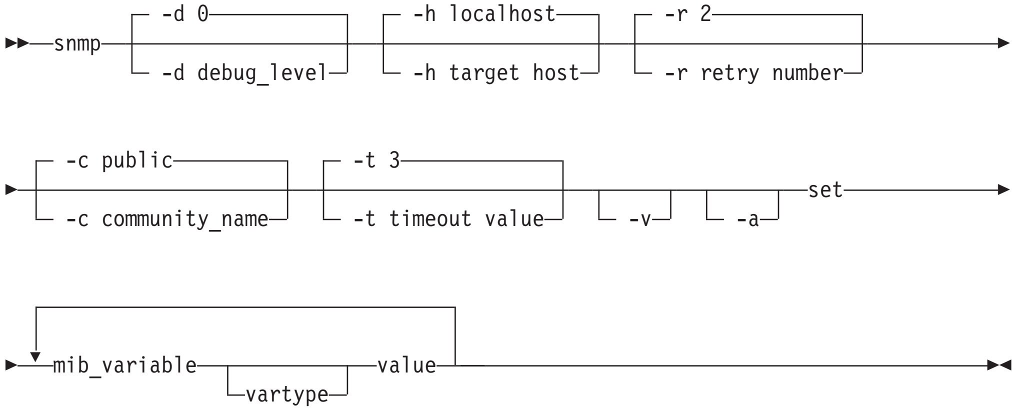
Note: **snmp** is a synonym for the **osnmp** command in the z/OS UNIX shell. The **osnmp** command syntax is the same as that for the **snmp** command.

Format

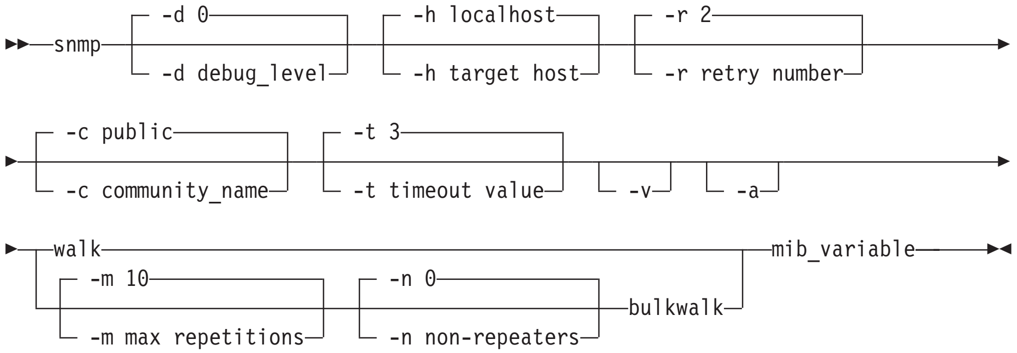
Getting MIB Variables:

```
          ┌─-d 0────────────┐  ┌─-h localhost──┐  ┌─-r 2──────────┐
►►──snmp──┼─────────────────┼──┼───────────────┼──┼───────────────┼──────────►
          └─-d debug_level──┘  └─-h target host┘  └─-r retry number┘

   ┌─-c public──────────┐  ┌─-t 3────────────┐
►──┼────────────────────┼──┼─────────────────┼──┬────┬──┬────┬───────────────►
   └─- c community_name─┘  └─-t timeout value┘  └─-v─┘  └─-a─┘

►──┬─get─────────────────────────────────────────────────────┬───────────────►
   ├─getnext─────────────────────────────────────────────────┤
   │ ┌─-m 10─────────────┐  ┌─-n 0─────────────┐              │
   └─┼───────────────────┼──┼──────────────────┼──getbulk─────┘
     └─-m max repetitions┘  └─-n non-repeaters─┘

   ┌───────────────┐
   ▼               │
►────mib_variable──┴──────────────────────────────────────────────────────►◄
```

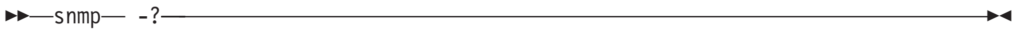
Setting the MIB Variables:



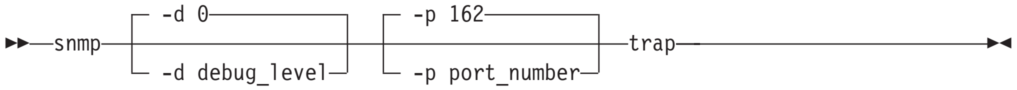
Walking the MIB Tree:



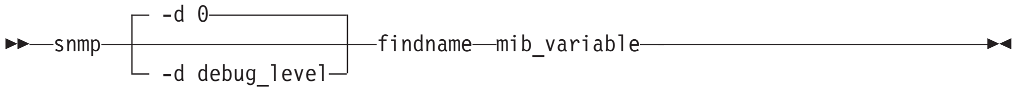
Displaying snmp Help:



Receiving a Trap:



Finding a MIB Variable Name:



Parameters

-d *debug_level*

Specifies the debug level. The valid debug levels are 0-4. The default level is 0, which means no debug. Each higher trace level includes all the trace messages from the lower levels.

-h *target host*
Specifies the target host to which you want to send a request. This can be an IPv4 (dotted decimal) or IPv6 (colon hexadecimal) address, a host name, or a winSNMP name in the OSNMP.CONF configuration file. If you do not specify a host, the default is your local host.

Restriction: You cannot specify scope information as part of the host name or the IP address of the target host.

-r *retry number*
Specifies the maximum number of times to retry the command if it timed out. The default is 2.

-c *community_name*
Specifies the community name that is used to access the specified variables at the destination SNMP agent. If you do not specify a community name, the default name is *public*. Community names are not required when using the user-based security model.

Note: Community names are case-sensitive.

-t *timeout value*
Specifies the amount of time (in seconds) that the **snmp** command waits for a reply from the SNMP agent. The default value is 3.

-v Specifies that the output from a request is displayed using verbose output. Use of this option causes the values to be returned with the textual name in place of the MIB object identifier.

-a Specifies that the request packet is sent using the physical interface addresses, rather than a VIPA address (if one is available) as the originating IP address. By default, the **snmp** command now uses the VIPA address. Alternately, the NOSVIPA option can be configured in the OSNMP.CONF file.

-m *max repetitions*
Applies only to getbulk and bulkwalk requests. This is ignored if the function request is not a getbulk or bulkwalk. Maximum repetitions is the number of lexicographic successors to be returned for each variable binding pair after the first -n number successors. For example, starting with successor -n number+1, return -m number of successors for each variable binding pair. The default is 10.

-n *nonrepeaters*
Applies only to getbulk or bulkwalk requests. This is ignored if the function request is not a getbulk or bulkwalk. The value *nonrepeaters* is the number of variable binding pairs (name and value), starting with the first, for which only a single successor is returned. The default value is 0.

mib_variable
Specifies the Management Information Base (MIB) object, using its object descriptor (textual name), object identifier in ASN.1 notation, or a combination of the two. When used with walk and bulkwalk requests, this is the MIB object prefix. A prefix can be any leading portion of the complete object identifier. When used with findname, this is the object identifier in ASN.1 notation.

vartype
Specifies the type of value being set. To complete an SNMP SET request, the SMI_type must be known. If no type is specified, **snmp** searches first the MIBS.DATA file and then the compiled MIB to determine the type. If the variable is not found, an error is returned. If a *vartype* is specified, the *vartype* takes precedence over any type that can be assigned in the MIB. The *vartype*

and value must be compatible. For example, if you specify a type of “number” and a *value* of “foo”, an error is returned because “foo” is not a number. The *vartype* parameter is not case-sensitive. Valid variable types are:

- bitstring
- counter
- counter32
- counter64
- display or displaystring
- integer
- integer32
- ipaddress
- gauge
- gauge32
- nsapaddress
- null
- objectidentifier or OID
- octetstring
- opaque
- opaqueascii
- timeticks
- uinteger

value
Specifies the value to be set by the SET function. If white space is needed in the value, you must enclose the value in double quotation marks ("). If you want to set a variable to a value that is also a type, you must specify the type.

-? Displays help information.

-p *port_number*
Specifies the number of the port that listens for traps. If a port number is not specified, the **snmp** command trap function listens on the well-known port 162, the default port for snmp traps.

SNMP request types:

get
Sends a request to an SNMP agent for a specific management information base (MIB) variable. The **snmp** command then waits for a response or times out.

getnext
Sends a request to an SNMP agent for the next MIB variable that lexicographically follows the *mib_variable* value specified. The **snmp** command then waits for a response or times out.

getbulk
Obtains the value of the variables in the MIB tree specified by the OID or MIB variable name. A single getbulk request performs the same function as a series of getnext requests with fewer data exchanges between the **snmp** command and the SNMP agent.

set
Sends a request to an SNMP agent to set a specific MIB variable. The **snmp** command then waits for a response or times out.

walk
Issues a getnext request for a specified prefix, then continues to issue getnext requests for as long as there are variables that match the specified prefix. A prefix can be any leading portion of the complete object identifier.

bulkwalk
Issues a GETBULK request for a specified prefix, then continues to issue GETBULK requests for as long as there are variables that match the specified prefix.

trap
Listens for SNMP traps and displays trap information when they occur. Uses the default well-known port 162 or the port number specified on the *-p* option. The **snmp** trap function continues to listen for traps until the process is killed or canceled.

findname
Sends a request that a search be done to obtain the textual name, for a given *mib_variable* input, whose internal ASN.1 value best matches the input ASN.1 value. The search first checks the MIBS.DATA file, and if a matching textual name is not found, continues with the compiled MIB. Only one *mib_variable* is allowed per **snmp** findname invocation.

Usage

- The set operation is not supported on all MIB objects. The set operation might be rejected if the agent or subagents managing the MIB object do not support SET.
- getbulk and bulkwalk are SNMPv2 functions. If the target agent supports only SNMPv1, the target agent ignores your request. As a result, your request times out.
- The function keywords are not case-sensitive. The - options and variable types and values are case-sensitive.
- In order to issue the **snmp** trap command, you must be in superuser mode if the use of the low port numbers is restricted by the UDPCONFIG statement in the TCP/IP profile. Low port access is required in order to bind to well-known port 162. If you are not in superuser mode, you receive error `EZZ3301I Error return from bind() : EDC5111I Permission denied`.

  For more information about the UDPCONFIG statement, see the z/OS Communications Server: IP Configuration Reference.
- In order to listen to traps from NetView SNMP and z/OS UNIX **snmp** command at the same time, use the *-p port_number* parameter on the **snmp** command. Only one management application at an IP address can listen on a port at a time. Specifying *-p* on the **snmp** trap command enables a port other than well-known port 162 to be used. Both ports must be configured as agent trap destinations.
- An **snmp** command that is not authenticated (by using an acceptable community name or user name) will time out.
- The **snmp** command uses two configuration files: MIBS.DATA and OSNMP.CONF. Sample files are shipped in the/usr/lpp/tcpip/samples directory. For information about these configuration files, see the z/OS Communications Server: IP Configuration Reference.
- The **snmp** command supports sending SNMPv1, SNMPv2c, and SNMPv3 requests. The snmp command uses the OSNMP.CONF file to determine whether it sends an SNMPv1, SNMPv2c, or SNMPv3 request. If the target specified by way of the *-h* parameter matches a winSNMP name in the OSNMP.CONF file, **snmp** sends the request using the parameters specified on the entry. If the *-h*

parameter is not specified, then the request will be sent as an SNMPv1 request. If the *-h* parameter is specified and is not found in the OSNMP.CONF file, the following error message is issued:

```
EZZ3306I Error converting <name> to Entity
```

Examples

- **Getting the MIB variable**

  The following requests MIB object sysName.0:

  ```
  snmp get sysName.0
  1.3.6.1.2.1.1.5.0 = MVS SNMP
  ```

  The following requests MIB object myName.0, where myName is defined in the MIBS.DATA file to be the same object identified by sysName.0:

  ```
  snmp get myName.0
  1.3.6.1.2.1.1.5.0 = MVSX SNMPv2 Agent
  ```

- **Getting the next MIB variable**

  The following requests the next logical MIB object:

  ```
  snmp getnext udp
  1.3.6.1.2.1.7.1.0 = 653
  ```

  The following requests the next logical object, using the *-v* option to have value displayed with textual name instead of object identifier:

  ```
  snmp -v getnext udp
  udpInDatagrams.0 = 653
  ```

- **Setting the MIB variable**

  The following sets MIB object sysName.0 to a value of ’MVSX SNMPv2 Agent’:

  ```
  snmp set sysName.0  "MVSX SNMPv2 Agent"
  1.3.6.1.2.1.1.5.0 = MVSX SNMPv2 Agent
  ```

  The following sets MIB object usmUserAuthKeyChange.1.2.2.117.49 to a hexadecimal value. Backward slashes are included in the value before each single quote to indicate to the UNIX shell that the single quote is part of the string to be passed to the **snmp** command. The 'h at the end of the value indicates that a hexadecimal value is passed.

  ```
  snmp set usmUserAuthKeyChange.1.2.2.117.49
  \'3eca6ff34b59010d262845210a40165678dd9646e31e9f890480a233dbe1114d\'h
  ```

- **Walking the MIB tree**

  The following returns by name all objects beginning with the same object identifier prefix:

```
snmp -v walk udp

udpInDatagrams.0 = 13
udpNoPorts.0 = 7
udpInErrors.0 = 0
udpOutDatagrams.0 = 20
udpLocalAddress.0.0.0.0.161 = 0.0.0.0
udpLocalAddress.0.0.0.0.514 = 0.0.0.0
udpLocalAddress.0.0.0.0.4001 = 0.0.0.0
udpLocalAddress.0.0.0.0.50003 = 0.0.0.0
udpLocalAddress.9.42.103.27.1029 = 9.42.103.27
udpLocalPort.0.0.0.0.161 = 161
udpLocalPort.0.0.0.0.514 = 514
udpLocalPort.0.0.0.0.4001 = 4001
udpLocalPort.0.0.0.0.50003 = 50003
udpLocalPort.9.42.103.27.1029 = 1029
udpEndpointProcess.0.0.161.0.0.0.33 = 0
udpEndpointProcess.0.0.514.0.0.0.28 = 0
udpEndpointProcess.0.0.4001.0.0.0.44 = 0
udpEndpointProcess.0.0.50003.0.0.0.70 = 0
udpEndpointProcess.1.4.9.42.103.27.1029.0.0.0.67 = 0
udpEndpointProcess.2.0.4002.2.16.32.1.13.184.0.0.0.0.0.0.0.0.0.0.0.1.9002.45 = 0
udpEndpointProcess.2.16.255.1.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.5003.0.0.0.46 = 0
```

- **Walking the tree using bulkwalk**

  The following returns by name all objects beginning with the same object identifier prefix, but with fewer data packages to be exchanged between the **snmp** command and the SNMP agent.

  The bulkwalk request type is an SNMPv2 function. The -h parameter identifies a host, loopback, defined in the OSNMP.CONF file as an agent that supports SNMPv2 or SNMPv3.

```
snmp -h loopback -v -m 10 bulkwalk udp

udpInDatagrams.0 = 2125
udpNoPorts.0 = 7
udpInErrors.0 = 0
udpOutDatagrams.0 = 2132
udpLocalAddress.0.0.0.0.161 = 0.0.0.0
udpLocalAddress.0.0.0.0.514 = 0.0.0.0
udpLocalAddress.0.0.0.0.4001 = 0.0.0.0
udpLocalAddress.0.0.0.0.50009 = 0.0.0.0
udpLocalAddress.9.42.103.27.1030 = 9.42.103.27
udpLocalPort.0.0.0.0.161 = 161
udpLocalPort.0.0.0.0.514 = 514
udpLocalPort.0.0.0.0.4001 = 4001
udpLocalPort.0.0.0.0.50009 = 50009
udpLocalPort.9.42.103.27.1030 = 1030
udpEndpointProcess.0.0.161.0.0.0.34 = 0
udpEndpointProcess.0.0.514.0.0.0.36 = 0
udpEndpointProcess.0.0.4001.0.0.0.44 = 0
udpEndpointProcess.0.0.50010.0.0.0.80 = 0
udpEndpointProcess.1.4.9.42.103.27.1031.0.0.0.78 = 0
udpEndpointProcess.2.0.4002.2.16.32.1.13.184.0.0.0.0.0.0.0.0.0.0.0.1.9002.45 = 0
udpEndpointProcess.2.16.255.1.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.5003.0.0.0.46 = 0
udpHCInDatagrams.0 = 2125
udpHCOutDatagrams.0 = 2132
```

- **Getting multiple MIB variables**

  The following requests multiple MIB objects using the getbulk request type. The getbulk request type returns the next logical object for one or more MIB objects listed on the command. In the following example, the *-n* option indicates that only one next logical object is requested for the first two variables (sysObjectId and ifNumber). For all other objects in the list (ifName, ifHCInOctets, ifHCOutOctets), the -m option indicates that 5 repetitions are requested. As a result of this command, the following SNMP data is returned:

- sysObjectId - the SNMP OID (object identifier) that identifies the agent
- ifNumber - the total number of interfaces defined to the TCP/IP stack with which the agent is associated
- The ifName, ifHCInOctets, and ifHCOutOctets values for the first 5 interfaces defined to the stack

The getbulk request type is an SNMPv2 function. The *-h* parameter identifies a host, loopback, defined in the OSNMP.CONF file as an agent that supports SNMPv2 or SNMPv3.

```
snmp -h loopback -v -n 2 -m 5 getbulk sysObjectId ifNumber ifName ifHCInOctets ifHCOutOctets

sysObjectID.0 = 1.3.6.1.4.1.2.3.13
ifNumber.0 = 31
ifName.1 = LOOPBACK
ifHCInOctets.1 = 108028
ifHCOutOctets.1 = 108028
ifName.2 = LOOPBACK
ifHCInOctets.2 = 107868
ifHCOutOctets.2 = 107868
ifName.3 = LOOPBACK6
ifHCInOctets.3 = 160
ifHCOutOctets.3 = 160
ifName.4 = LCS1
ifHCInOctets.4 = 0
ifHCOutOctets.4 = 0
ifName.5 = TR1
ifHCInOctets.5 = 0
ifHCOutOctets.5 = 0
```

- **Finding the name of an ASN.1 variable**

  The following sends a request that a search be done to obtain the textual name, for a given *mib_variable* input, whose internal ASN.1 value best matches the input ASN.1 value. The search begins with the MIBS.DATA file and, if not found, continues with the compiled MIB. Only one *mib_variable* is allowed per **snmp** findname command invocation:

```
snmp findname 1.3.6.1.2.1.6.13.1.2
1.3.6.1.2.1.6.13.1.2 found as: tcpConnLocalAddress
```

```
snmp findname 1.3.6.1.2.1.6.13.1.2.0
1.3.6.1.2.1.6.13.1.2.0 found as: tcpConnLocalAddress.0
```

```
snmp findname 1.3.6.1.2.
1.3.6.1.2. found as: mgmt
```

- **Sending requests with the physical interface address as originating address:**

  By default, the **snmp** command no longer sets the SO_IGNORESOURCEVIPA socket option to force the originating address in the request packet to be that of the physical interface over which the packet is sent. A source VIPA address, if one is configured, is used instead. To cause the **snmp** command to use the physical address instead, the -a option can be specified. This implies that the SNMP agent receiving the request must be configured to accept requests from the physical interface address rather than the source VIPA address.

  To have the **snmp** command use the physical interface address as the originating address, use the -a parameter on the **snmp** command:

```
   snmp -a get sysUpTime.0
   1.3.6.1.2.1.1.3.0 = 2950600
```

Alternately, if an entry exists in the OSNMP.CONF file for hostA that specifies NOSVIPA, the following command would achieve the same results:

```
snmp -h hostA get sysUpTime.0
1.3.6.1.2.1.1.3.0 = 2950600
```

Using SNMP from NetView

If you want to use SNMP from NetView, you have several alternatives. The most basic is the command line interface, the NetView SNMP command, documented in "The NetView SNMP command."

For more sophisticated management support, consider using the AON support provided in *Tivoli NetView for z/OS*, which provides panel-based support for retrieval and modification of SNMP management data at a TCP/IP host. For additional information, see the *Tivoli NetView for z/OS Automated Operations Network User's Guide*, GC31-8851.

Additionally, z/OS Communications Server provides sample NetView command lists. These are sample files only and do not reflect the most recent MIB variable support. Two sets of sample command lists are provided to execute SNMP requests from full-screen mode. One, written in the NetView Command List (CLIST) language, is documented in the SNMPCLST.README file. The other is written in REXX and is documented in the SNMPREXX.README file.

The NetView SNMP command

To issue an SNMP request from NetView, use the SNMP command. The SNMP command provides SNMP manager function with the NetView program to query SNMP agents for network management information.

The NetView SNMP command uses the SNMP Query Engine to issue SNMP requests to agents and to process SNMP responses returned by agents. The SNMP command supports issuance of SNMPv1 requests.

The SNMP command does not support the use of IPv6 addresses.

Note: The z/OS Communications Server SNMP agent supports SNMPv1, SNMPv2c, and SNMPv3 requests.

Format

Getting MIB Variables:

```
                                                    ┌──────────┐
                                                    ▼          │
►►──SNMP──┬─Get─────┬──host_name──community_name──────var_name─┴──────────────►◄
          └─GETNext─┘
```

Setting the MIB Variables:

```
                                           ┌──────────────────┐
                                           ▼                  │
►►──SNMP Set──host_name──community_name──────var_name──value──┴───────────────►◄
```

Finding an ASN.1 Variable Name:

```
►►──SNMP MIBvname──asn.1 name──────────────────────────────────────────────►◄
```

Forwarding Traps:

```
►►──SNMP TRAPson──net_mask──net_desired────────────────────────────────────►◄
```

Stop Forwarding Traps:

```
►►──SNMP TRAPSOFf──filter_id───────────────────────────────────────────────►◄
```

Pinging a Node:

```
►►──SNMP PING──host_name───────────────────────────────────────────────────►◄
```

Parameters

SNMP request types

Get
Sends a request to an SNMP agent for a specific management information base (MIB) variable.

GETNext
Sends a request to an SNMP agent for the next MIB variable that lexicographically follows the *var_name* specified.

Set
Sends a request to an SNMP agent to set a specific MIB variable.

MIBvname
Requests the textual name of an ASN.1 MIB object.

TRAPson
Requests that the SNMP Query Engine listen on the trap port for SNMP traps and forward them to the NetView program, which displays trap information when it occurs.

TRAPSOFf
Causes the SNMP Query Engine to stop listening on the trap port for SNMP traps and stop forwarding them to the NetView program.

PING
Obtains the minimum round-trip response time from the Query Engine to a specific node.

Variables

host_name
Specifies the destination host to which you want to send a request. The host can be specified with its name or with its IP address in dotted decimal notation.

community_name
Specifies the community name used to access the specified variables at the destination SNMP agent.

Note: Community names are case-sensitive. SNMP commands issued from the NetView console are converted to uppercase. Those issued from REXX execs are not converted to uppercase.

var_name
Specifies one or more MIB variable names to be retrieved or set. You can specify the textual names or ASN.1 notation (for example, sysDescr.0 or 1.3.6.1.2.1.1.1.0). The SNMP Query Engine can accept a maximum of 10 variables for each request.

All MIB variables that are defined as part of a sequence represent variables that can have more than one occurrence. These variables require an instance identifier appended to the end of the variable name to identify which occurrence of the variable is being requested.

value
Specifies the value to be set by the SET function. On the Set command from the NetView console, a value is enclosed in single quotation marks, not double quotation marks. From the panels, you can specify no quotation marks, single quotation marks (′), or double quotation marks (″). No quotation marks and single quotation marks work the same. If you specify double quotation marks, you get double quotation marks as part of the value.

asn.1_name
Specifies the MIB object, using its object identifier in ASN.1 notation. You can specify only one variable. Additional arguments are ignored.

net_mask
Specifies, in dotted decimal notation, the network mask to be evaluated with the IP address of incoming traps. The dotted decimal IP address is ANDed with this mask.

net_desired
Specifies the network from which you want to receive traps.

filter_id
Specifies the trap filter ID.

When you request traps using the SNMP TRAPSON command, it returns a request number or *filter_id*, which the SNMP Query Engine associates with the TRAPSON request. To stop receiving traps, specify this *filter_id* in the TRAPSOFF request.

Usage

- If you start and stop NetView, you must do the same to the SNMP Query Engine.
- When the SNMP command is issued from the NetView Command Facility command line, all input is translated to uppercase (standard NetView format) before it is sent to the SNMP Query Engine.
- When the SNMP command is issued from a CLIST, input is passed in whatever case it was passed from the CLIST (for example, mixed case).
- The textual names for the variables passed to the query engine are compared against the entries in the MIBDESC.DATA file. This comparison is not case-sensitive.
- If multiple variables are specified with the GET, GETNext, or SET commands, they are all packaged in one SNMP PDU to be sent to the agent.

- If multiple SNMP requests are issued, the responses might not be received in the same order the requests are issued.
- The SNMP agent can receive SNMP requests over any interface.
- The SNMP Query Engine treats numbers with leading zeros as octal numbers. Therefore, do not use leading zeros.
- If an SNMP request is issued with the wrong community name, it could receive multiple AUTHENTICATION FAILURE traps with the same *filter_id* but different time stamps from the same host. This is because the SNMP Query Engine tries the request again if a response is not received from the host, and each attempt causes the host to generate an AUTHENTICATION FAILURE trap.

Return codes

The following table lists the return codes generated by SNMP.

Return code	Description
1	Error from DSIGET, cannot continue
2	Incorrect function specified
3	Missing SNMP function
4	Not enough parameters
5	Missing variable name
6	Missing variable value
7	Missing or incorrect host name
8	Missing community name
9	SNMPIUCV not active
10	Error from DSIMQS
11	Incorrect *net_mask*/desired network
12	Missing/Incorrect trap *filter_id*
1001+	Command successful — all return codes above 1000

Examples

- **Retrieving the MIB variable**

  For example, if you know:

```
hostname           -  anyhost
IP address         -  129.34.222.72
community name     -  public
variable name      -  sysDescr.0
asn.1 variable name -  1.3.6.1.2.1.1.1.0
variable name      -  sysObjectID.0
asn.1 variable name -  1.3.6.1.2.1.1.2.0
variable name      -  sysUpTime.0
asn.1 variable name -  1.3.6.1.2.1.1.3.0
```

  You can issue the following SNMP GET commands:

```
snmp get 129.34.222.72 public 1.3.6.1.2.1.1.1.0
snmp get 129.34.222.72 public sysDescr.0
snmp get anyhost public 1.3.6.1.2.1.1.1.0
snmp get anyhost public sysDescr.0
snmp get anyhost public sysObjectID.0
snmp get anyhost public sysUpTime.0
snmp get anyhost public sysDescr.0 sysObjectID.0 sysUpTime.0
```

After the last SNMP GET command is completed, you get a message similar to the following one:

```
SNM050I SNMP Request 1001 from NETOP accepted, sent to Query Engine
```

When the response arrives in the NetView program (asynchronously), it displays the response as a multiline message in the following form:

```
SNM040I SNMP Request 1001 from NETOP Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.1.0
SNM043I Variable value type: 9
SNM044I Variable value: AIX 2.2.1 SNMP Agent Version 1.0
SNM042I Variable name: 1.3.6.1.2.1.1.2.0
SNM043I Variable value type: 3
SNM044I Variable value: 1.3.6.1.4.1.2.1.1
SNM042I Variable name: 1.3.6.1.2.1.1.3.0
SNM043I Variable value type: 8
SNM044I Variable value: 98800
SNM049I SNMP Request 1001 end of response
```

Figure 4. SNMP request response

- **Retrieving the next MIB variable**

  For example, if you know:

  ```
  hostname           -  anyhost
  IP address         -  129.34.222.72
  community name     -  public
  variable name      -  ifAdminStatus (in ifTable)
  asn.1 variable name -  1.3.6.1.2.1.2.2.1.7
  ```

  You can issue an SNMP GETNext command in one of the following ways:

  ```
  snmp getnext 129.34.222.72 public 1.3.6.1.2.1.2.2.1.7.0
  snmp getnext 129.34.222.72 public ifAdminStatus.0
  snmp getnext anyhost public 1.3.6.1.2.1.2.2.1.7.0
  snmp getnext anyhost public ifAdminStatus.0
  ```

  The GETNext command is completed in the same manner as the GET command, and you receive an asynchronous response similar to the following one:

  ```
  SNM040I SNMP Request 1001 from NETOP Returned the following response:
  SNM042I Variable name: 1.3.6.1.2.1.2.2.1.7.1
  SNM043I Variable value type: 1
  SNM044I Variable value: 1
  SNM049I SNMP Request 1001 end of response
  ```

  In this example, the first instance of the variable has a status of 1 or greater (ends in `7.1`).

  You can then issue another GETNext command in one of the following ways:

  ```
  snmp getnext 129.34.222.72 public 1.3.6.1.2.1.2.2.1.7.1
  snmp getnext 129.34.222.72 public ifAdminStatus.1
  snmp getnext anyhost public 1.3.6.1.2.1.2.2.1.7.1
  snmp getnext anyhost public ifAdminStatus.1
  ```

  The GETNext command is completed in the same manner as the GET command, and you receive an asynchronous response similar to the following one:

  ```
  SNM040I SNMP Request 1002 from NETOP Returned the following response:
  SNM042I Variable name: 1.3.6.1.2.1.2.2.1.7.2
  SNM043I Variable value type: 1
  SNM044I Variable value: 1
  SNM049I SNMP Request 1002 end of response
  ```

  In this example, the second instance of the variable has a status of 1 or greater (ends in `7.2`).

  You can then issue another GETNext command in one of the following ways:

```
snmp getnext 129.34.222.72 public 1.3.6.1.2.1.2.2.1.7.2
snmp getnext 129.34.222.72 public ifAdminStatus.2
snmp getnext anyhost public 1.3.6.1.2.1.2.2.1.7.2
snmp getnext anyhost public ifAdminStatus.2
```

The GETNext command is completed in the same manner as the GET command, and you receive an asynchronous response similar to the following one:

```
SNM040I SNMP Request 1003 from NETOP Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.2.2.1.8.1
SNM043I Variable value type: 1
SNM044I Variable value: 1
SNM049I SNMP Request 1003 end of response
```

- **Setting the MIB variable**

  For example, if you know:

```
hostname           -  anyhost
IP address         -  129.34.222.72
community name     -  publicw
variable name      -  ifAdminStatus
asn.1 variable name -  1.3.6.1.2.1.2.2.1.7.1
     (instance 1)
```

  You can then issue an SNMP SET command in one of the following forms to set the administrative status of the first interface in the ifTable (first instance) to test:

```
snmp set 129.34.222.72 publicw 1.3.6.1.2.1.2.2.1.7.1 3
snmp set 129.34.222.72 publicw IfAdminStatus.1 3
snmp set anyhost publicw 1.3.6.1.2.1.2.2.1.7.1 3
snmp set anyhost publicw ifAdminStatus.1 3
```

  After the command is completed, you receive a message similar to the following one:

```
SNM050I SNMP Request 1001 from NETOP accepted, sent to Query Engine
```

  When the response arrives in the NetView program (asynchronously), it displays the response as a multiline message in the following form:

```
SNM040I SNMP Request 1001 from NETOP Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.2.7.1
SNM043I Variable value type: 1
SNM044I Variable value: 3
SNM049I SNMP Request 1001 end of response
```

  If a SET request is attempted against an object for which the target agent or subagent does not allow SETs, you receive:

  – `noSuchName for SNMPv1 requests`

  Appendix B, “Management Information Base (MIB) objects,” on page 1025 identifies the objects supported by the z/OS Communications Server SNMP agent and subagents and the level of access supported for each object.

  Note: The variable being set must be present in the MIBDESC.DATA data set for the Query Engine to determine the syntax to use when encoding the SNMP PDU.

- **Receiving a trap**

  The SNMP TRAPSON command permits the specification of a filtering condition that enables the Query Engine to perform filtering. The SNMP TRAPSON command assigns a unique request number to each filter (also called a *filter_id*) and returns this number in a message and in the return code. This *filter_id* is the argument to an SNMP TRAPSOFF command, which is used to stop receiving traps that pass this filter.

For example, if you know:

```
IP address          -  129.34.222.72
net mask            -  255.255.255.255
```

You can issue the following SNMP TRAPSON commands:

```
snmp trapson
snmp trapson 255.255.255.255 129.34.222.72
```

The first command receives all traps (the default is a mask of 0 and a required network of 0). The second command receives traps only from the specific host 129.34.222.72.

After the command is completed, you receive a message similar to the following one:

```
SNM050I SNMP Request 1001 from NETOP accepted, sent to Query Engine
```

The number returned in the message (1001 in the previous example) is used as the *filter_id*. This *filter_id* is displayed in the header message of traps passed by this filter. The *filter_id* is used in the TRAPSOFF command to turn the filter off.

When the response arrives in the NetView program (asynchronously), it displays the response as a multiline message in the following form to indicate that the TRAPSON request was accepted:

```
SNM040I SNMP Request 1001 from NETOP Returned the following response:
SNM045I Major error code: 0
SNM046I Minor error code: 0
SNM047I Error index: 0
SNM048I Error text: no error
SNM049I SNMP Request 1001 end of response
```

When traps arrive, the NetView program displays each trap with a multiline message in the following form. This multiline message is sent to the NetView operator who is designated as the authorized receiver (AUTH MSGRECVR=YES in the operator profile); it might not show up on the console of the operator who issues the TRAPSON command.

```
SNM030I SNMP request 1001 received following trap:
SNM031I Agent Address: 129.34.222.34
SNM032I Generic trap type: 4
SNM033I Specific trap type: 0
SNM034I Time stamp: 472600
SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.1.1
SNM039I SNMP request 1001 End of trap data
```

After the TRAPSON command has been issued, traps can start to arrive asynchronously. They can even arrive after the operator who issued the TRAPSON command has logged off. Often, a TRAPSON command is issued by a CLIST, and the received trap data triggers another CLIST to handle the trap data. Therefore, the messages in the range SNM030—SNM039 are sent to the authorized receiver. For a NetView operator to see the traps, the operator must have the following statement in the NetView Operator profile:

```
AUTH MSGRECVR=YES
```

However, only one operator receives the message. The message also goes to the log file, so you can always browse the log file to see trap data. Additionally, you can assign trap messages to go to a specific operator using the NetView ASSIGN operator command.

In the response to the SNMP TRAPSON request, not all lines need to be present, but the first line is always message SNM040I, and the last line is always message SNM049I.

For the multiline trap message, not all lines need to be present, but the first line is always message SNM030I, and the last line is always message SNM039I.

Additional messages (SNM036I—SNM038I) could be present if the trap has additional data.

If a variable value is too long, message SNM038 might not fit on an 80-character line. If this happens, the value is split and multiple SNM038 messages are displayed.

The SNMP trap data always displays the variable name in ASN.1 notation. You can use SNMP MIBVNAME to obtain the textual name for the variable.

A trap always shows the agent address in the form of an IP address in dotted decimal notation.

You can issue multiple TRAPSON requests, with either the same or a different filter. If a trap passes multiple filters, the trap is sent to the NetView program multiple times. However, in the NetView program, the header and trailer lines (messages SNM030I and SNM039I) of the duplicate trap are different, because they contain the *filter_id* (request number) by which the trap was forwarded. Different types of traps from different hosts can have the same *filter_id*, if these traps pass the same trap filter.

The SNMP Query Engine can forward only those traps that it receives. Each agent has a trap destination table, which lists all the hosts that receive that agent's traps. The host name of your system must be in the trap destination table of all agents from which you want to receive traps.

- **Stop listening for traps**

  For example, if you know the *filter_id* is 1001, you can issue the following SNMP TRAPSOFF command to tell the SNMP Query Engine to quit sending traps that would pass filter 1001:

  ```
  snmp trapsoff 1001
  ```

  The command completes with a message similar to the following one:

  ```
  SNM050I SNMP Request 1001 from NETOP accepted, sent to Query Engine
  ```

  When the response arrives in the NetView program (asynchronously), it displays the response as a multiline message in the following form to indicate that the TRAPSOFF request was accepted.

  ```
  SNM040I SNMP Request 1002 from NETOP Returned the following response:
  SNM045I Major error code: 0
  SNM046I Minor error code: 0
  SNM047I Error index: 0
  SNM048I Error text: no error
  SNM049I SNMP Request 1002 end of response
  ```

  Only one *filter_id* for each SNMP TRAPSOFF command can be passed. Extraneous arguments are ignored.

- **Finding the name of an ASN.1 variable**

  For example, if you have a trap that tells you:

```
SNM030I SNMP request 1001 received following trap:
SNM031I Agent Address: 129.34.222.34
SNM032I Generic trap type: 2
SNM033I Specific trap type: 0
SNM034I Time stamp: 472600
SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.1.1
SNM036I Variable name: 1.3.6.1.2.1.2.2.1.1
SNM037I Variable value type: 1
SNM038I Variable value: 2
SNM039I SNMP request 1001 End of trap data
```

You can issue the following SNMP MIBVNAME command to find the textual MIB variable name:

```
snmp mibvname 1.3.6.1.2.1.2.2.1.1
```

The command completes with a message similar to the following one:

```
SNM050I SNMP Request 1002 from NETOP accepted, sent to Query Engine
```

When the response arrives in the NetView program (asynchronously), it displays the response as a multiline message in the following form:

```
SNM040I SNMP Request 1002 from NETOP Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.2.2.1.1
SNM043I Variable value type: 9
SNM044I Variable value: ifIndex
SNM049I SNMP Request 1002 end of response
```

Only one ASN.1 variable name can be passed for each SNMP MIBVNAME command. Additional parameters are ignored.

- **Pinging a node**

For example, if you know:

```
nodename           -  anynode
IP address         -  129.34.222.72
```

You can issue the following SNMP PING commands:

```
SNMP PING ANYNODE
SNMP PING 129.34.222.72
```

The command completes with a message similar to the following one:

```
SNM050I SNMP Request 1001 from NETOP accepted, sent to Query Engine
```

When the response arrives in the NetView program (asynchronously), it displays the response as a multiline message in the following form:

```
SNM040I SNMP Request 1001 from NETOP Returned the following response:
SNM042I Variable name: 1.3.6.1.4.1.2.2.1.3.2.129.34.222.72
SNM043I Variable value type: 1
SNM044I Variable value: 26
SNM049I SNMP Request 1001 end of response
```

The Query Engine issues one PING (an ICMP echo on a raw socket) and returns the value in milliseconds in an IBM-defined SNMP variable minRTT. Because only one PING is issued, this is also the average and the maximum response time.

If the PING does not respond, the Query Engine tries again twice, once after one second and again after two seconds (Query Engine default retry mechanism). If a response is not received after all retries have been exhausted, a variable value of `-1` is returned to indicate that a reply was not received.

The 129.34.222.72 in the example for the SNMP PING command represents an instance of the IBM variable minRTT.

Only one node name can be passed for each SNMP PING command.

SNMP uses ICMP Echo to send a PING command to the remote host. No SNMP PDU exchange with the remote host occurs. Therefore, a successful SNMP PING indicates only that the remote host is active and reachable. It does not indicate that the SNMP agent at the remote host is active, or that the SNMP manager can send requests to the SNMP agent if it is active.

Usage

- The SNMP response always displays the variable name in ASN.1 notation. You can use SNMP MIBVNAME to obtain the textual name for the variable.
- If you issue a GET for multiple variables, messages SNM042—SNM044 are displayed for each variable.
- When you issue a GET for multiple variables, they are returned in the same sequence as requested. In Figure 4 on page 965, GET was issued for `sysDescr.0 sysObjectID.0 sysUpTime.0`. The same 3 variables are returned in the response.
- If an error was detected, messages SNM042–SNM044 might not be present. You can get (in addition to other messages) error messages in the following forms (all as part of multiline message SNM040I):

```
SNM045I Major error code: n
SNM046I Minor error code: y
SNM047I Error index: z
SNM048I Error text: message text
```

- If a variable value is too long, message SNM044 might not fit on an 80-character line. If this happens, the value is split and multiple SNM044 messages are displayed.
- According to RFC 1157 (*Simple Network Management Protocol (SNMP)*), a message exchanged between SNMP entities (including version identification and community name) can be as small as 484 octets. If you specify up to 10 variables in a GET/GETNext command, the names could be short enough to send the GET command to the SNMP agent, but the response could be too long to fit in the message. As a result, you receive a `tooBig` error.
- If one (or more) of the variables requested results in an error, all variables listed after the first variable in error are ignored, and data is not returned for them.
- To correctly retrieve the next variable for the GETNext command, you must specify an instance identifier as part of the variable name. If the variable has only one occurrence, or if the first occurrence of a table variable is needed, use .0 as the instance identifier.
- The GETNext command is used to interrogate a table (for example, the interface table) or an array. You can issue a GETNext command at the start of a table (use instance 0.0). The first element in the table is returned. The process continues in a loop, performing GETNext requests on the previously obtained variable name, until the name of the variable returned no longer has the same prefix as the one at the start of the table. This condition occurs when the GETNext request returns a variable that is in the next group.

Context

For information about the variable ibmMvsRPingResponseTime, which enables you to send remote PING commands, see “SNMP remote PING” on page 982.

For a list of variables supported by the z/OS Communications Server IP agent, see Appendix B, “Management Information Base (MIB) objects,” on page 1025.

Host name resolution

When a NetView SNMP request uses a symbolic host name rather than an IP address, the SNMP Query Engine uses the standard gethostbyname() function to look up the IP address of that host. The IP address is then saved in an SNMP Query Engine in-memory cache for future reference. Use of this cache improves the performance of subsequent requests for the same host.

Note: Because the cache cannot be refreshed, if the mapping between host names and IP addresses changes, you must restart the SNMP Query Engine (the SQESERV module) to rebuild the cache. You must also restart the SNMP Query Engine after a host name is added to the name server data base.

Major and minor error codes and SNMP value types

This section describes the possible major and minor error codes and variable value types that can be returned in a NetView SNMP response or trap.

- The major error code can have one of the following values:

Value	Major error code
0	No error detected
1	SNMP agent reported error
2	Internally detected error

- The minor error code can have one of the following values when the major error code indicates that an SNMP agent detected an error (1):

Value	SNMP Agent detected minor error code
0	No error
1	Too big
2	No such name
3	Incorrect value
4	Read only
5	General error

- The minor error code can have one of the following values when the major error code indicates that an internal error was detected (2):

Value	Internal minor error code
0	No error
1	Protocol error
2	Out of memory
3	No response–all retries failed
4	Some I/O error occurred
5	Illegal request
6	Unknown host specified
7	Unknown MIB variable
8	No such filter
9	Too many variables specified

- If the major error code indicates that an SNMP agent detected the error (1), the error index indicates the position of the first variable in error.
- The variable value type is one of the following, as specified in RFC 1155 (*Structure and Identification of Management Information for TCP/IP-Based Internets*) and RFC 1156 (*Management Information Base for Network Management of TCP/IP-Based Internets*):

Value	Value type
0	Text representation
1	Number (integer, signed)
2	Binary data string
3	Object identifier
4	Empty (no value)
5	IP address
6	Counter (unsigned)
7	Gauge (unsigned)
8	Time ticks (1/100ths seconds)
9	Display string

Note: The binary data string is displayed in the NetView program as a contiguous string of hexadecimal characters (for example, X'0123' is displayed as `0123`).

Creating user keys

The following information describes authentication keys and z/OS Communications Server encryptions support.

Authentication

Authentication is generally required for SNMPv3 requests to be processed (unless the security level requested is ’noAuth’). When authenticating a request, the SNMP agent verifies that the authentication key sent in an SNMPv3 request can be used to create a message digest that matches the message digest created from the authentication key defined for the user.

The **snmp** command uses the authentication key found on an entry in the OSNMP.CONF configuration file. It needs to correlate with the authentication key specified on a USM_USER entry for that user in the agent SNMPD.CONF configuration file.

As an alternative to storing authentication keys in the client configuration file, the **snmp** command allows user passwords to be stored. If the **snmp** command is configured with a password, the code will generate an authentication key (and privacy key if requested) for the user. These keys must, of course, produce the same authentication values as the keys configured for the USM_USER in the agent's SNMPD.CONF file or configured dynamically with SNMP SET commands. Note, however, the use of passwords in the client configuration file is considered less secure than the use of keys in the configuration file.

The authentication key is generated from two pieces of information:

- The specified password

- The identification of the SNMP agent at which the key will be used. If the agent is an IBM agent and its engineID was generated using the vendor-specific engineID formula, the agent might be identified by IP address or host name. Otherwise, the engineID must be provided as the agent identification.

A key that incorporates the identification of the agent at which it will be used is called a localized key. It can be used only at that agent. A key that does not incorporate the engineID of the agent at which it will be used is called nonlocalized.

Keys stored in the **snmp** command configuration file, OSNMP.CONF, are expected to be nonlocalized keys. Keys stored in the SNMP agent's configuration file, SNMPD.CONF, can be either localized or nonlocalized, though the use of localized keys is considered more secure.

Encryption

As of z/OS V1R2 Communications Server, encryption support is provided in the base product. Keys used for encryption are generated using the same algorithms as are used for authentication. However, key lengths might differ. For example, an HMAC-SHA authentication key is 20 bytes long, but a localized encryption key used with HMAC-SHA are only 16 bytes in length.

Using the pwtokey facility

z/OS Communications Server provides a facility called pwtokey that allows conversion of passwords into localized and non-localized authentication and privacy keys, for SNMP or OMPROUTE.

- For OMPROUTE, pwtokey takes as input a password and generates an authentication key. No localized or privacy keys are needed or generated for OMPROUTE. Some restrictions apply when using pwtokey for OMPROUTE. See the description of the password parameter for more information.
- For SNMP, the pwtokey procedure takes as input a password and an identifier of the agent and generates authentication and privacy keys. The procedure used by the pwtokey facility is the same algorithm used by the z/OS UNIX **snmp** command. The person configuring the SNMP agent can generate appropriate authentication and privacy keys to put in the SNMPD.CONF file for a user, given a particular password and the IP address at which the agent runs.

  Tip: For privacy, encryption requires the use of keys of 32 hexadecimal digits (16 bytes) in length. However, if the key is generated by using HMAC-SHA, which produces keys of 40 hexadecimal digits (20 bytes) in length, the truncation from 40 to 32 hexadecimal digits is not done until after the key is localized. Therefore, a non-localized privacy key generated using HMAC-SHA is 40 hexadecimal digits (20 bytes) long, and a localized privacy key generated using HMAC-SHA is 32 hexadecimal digits (16 bytes) long. A privacy key generated with HMAC-MD5 (localized or not) is 32 hexadecimal digits (16 bytes) long.

To convert passwords into authentication and privacy keys, issue the following command from z/OS UNIX to use the pwtokey facility.

Format

```
                      ┌─-d──0─┐  ┌─-p HMAC-MD5─┐  ┌─-u auth──────┐
►►──pwtokey──┬──────┬─┴───────┴──┼─────────────┼──┴──────────────┴──┬────┬──►
             └─-e───┘ └─-d──n─┘  ├─-p HMAC-SHA─┤  └─-u key_usage─┘  └─-s─┘
                                 └─-p all──────┘
```

```
►─password──┬───────────┬──────────────────────────────────────────────►◄
            ├─IPaddress─┤
            ├─hostname──┤
            └─engineID──┘
```

Parameters

-e This flag indicates that the agent for which the key is being defined is identified by engineID rather than by IP address or host name. This is applicable only when generating keys for SNMP.

-d *n*
This flag indicates what level of debug information is wanted. Debug tracing is either on or off, so a value of 1 causes debug tracing to be generated to the screen of the command issuer (sysout), and a value of 0 specifies that no debug tracing be generated. Debug tracing is off (0) by default.

-p *protocol*
This flag indicates the protocols for which the keys are generated. Valid values are:

HMAC-MD5
Generates keys for use with the HMAC-MD5 authentication protocol. This is the only protocol to use when generating OSPF MD5 keys for OMPROUTE.

HMAC-SHA
Generates keys for use with the HMAC-SHA authentication protocol.

all Generates both HMAC-MD5 and HMAC-SHA keys.

The default is that keys for the HMAC-MD5 protocol are generated.

-u *key_usage*
This flag indicates the usage intended for the key. Valid values are:

auth An authentication key. This is the recommended usage for generating OSPF MD5 keys for OMPROUTE.

priv A privacy key.

all Both authentication and privacy keys.

Note: There is no difference between a key generated for authentication and a key generated for privacy. However, the length of privacy keys depends on whether the key is localized or not.

-s This flag indicates that output data is displayed with additional spaces to improve readability. By default, data is displayed in a condensed format to facilitate cut-and-paste operations on the keys into configuration files or command lines.

password
Specifies the text string to be used in generating the keys. The *password* must be in the range of 8–255 characters long. In general, while any printable characters can be used in the passwords, the z/OS UNIX shell might interpret some characters rather than passing them to the pwtokey command. Include passwords in single quotation marks to avoid interpretation of the characters by the z/OS UNIX shell.

Note:

1. This password is not related to the community name (or password) used with community-based security (SNMPv1 and SNMPv2c). This password is used only to generate keys for user-based security, an entirely different security scheme.
2. For easier OMPROUTE migration from password to MD5 authentication, you can base the input password on the OMPROUTE password (there is no requirement for you to do so). Because the input password must be at least 8 characters and OMPROUTE supports passwords as few as 1 character, it might be necessary for you to pad or otherwise alter the OMPROUTE password to bring it up to 8 characters. Some restrictions apply when using PWTOKEY for OMPROUTE. See the MD5 Authentication specification for OMPROUTE in the z/OS Communications Server: IP Configuration Reference.

IPaddress
Specifies the IP address in IPv4 dotted decimal or IPv6 colon hexadecimal notation of the SNMP agent at which the key will be used on an SNMP request. This parameter is used only in generation of the localized key, and is not needed when generating MD5 keys for OMPROUTE.

hostname
Specifies the SNMP agent at which the key will be used on an SNMP request. This parameter is used only in generation of the localized key and is not needed when generating MD5 keys for OMPROUTE.

engineID
Specifies the engine ID of the SNMP agent at which the key will be used. The engine ID is determined at SNMP agent initialization from the SNMPD.BOOTS file. The engine ID must be a string of 1–32 octets (2–64 hexadecimal digits). If the engineID is specified, the -e option must also be specified. The default is that the agent identification is not an engine ID. This parameter is used only in generation of the localized key and is not needed when generating MD5 keys for OMPROUTE.

Examples

Sample output from the **pwtokey** command:

```
# pwtokey testpassword 9.67.113.79
Display of 16 byte HMAC-MD5 authKey:
 775b109f79a6b71f94cca5d22451cc0e

Display of 16 byte HMAC-MD5 localized authKey:
 de25243d5c2765f0ce273e4bcf941701
```

pwtokey generates two keys – one that is localized (has been tailored to be usable only at the agent identified) and one that has not been localized. Typically, the localized key is used in the configuration for the SNMP agent. The nonlocalized key is used in the configuration for the **snmp** command.

If pwtokey is invoked requesting HMAC-SHA keys for both authentication and privacy, the output looks like this:

```
# pwtokey -p HMAC-SHA -u all testpassword 9.67.113.79
Display of 20 byte HMAC-SHA authKey:
 b267809aee4b8ef450a7872d6e348713f04b9c50

Display of 20 byte HMAC-SHA localized authKey:
 e5438092d1098a43e27e507e50d32c0edaa39b7c

Display of 20 byte HMAC-SHA privKey:
 b267809aee4b8ef450a7872d6e348713f04b9c50

Display of 16 byte HMAC-SHA localized privKey:
 e5438092d1098a43e27e507e50d32c0e
```

The output for the privacy keys is the same as the output for the authentication keys, except that the localized privacy key has been truncated to 16 bytes as is required.

Note: If encryption is used, it is more secure to use different passwords for authentication and privacy.

If pwtokey is invoked requesting an MD5 authentication key for OMPROUTE, the output looks like this:

```
# pwtokey testpassword
Display of 16 byte HMAC-MD5 authKey:
 775b109f79a6b71f94cca5d22451cc0e
```

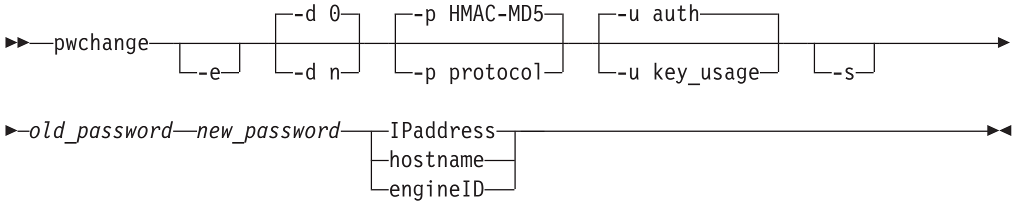
Usage

If the IP address or the host name is specified, the SNMP agent must be an IBM agent. The engineID is created using a vendor-specific formula that incorporates the IP address of the agent and an Enterprise ID representing IBM.

Using the pwchange facility

The pwchange command is provided to facilitate dynamic changes of user authentication and privacy keys. Dynamic configuration of authentication and privacy keys is done by doing SET commands to objects of syntax keyChange. The keyChange syntax provides a way of changing keys without requiring that the actual keys (either new or old) be flowed directly across the wire, which would not be secure. Instead, if an object, such as usmUserAuthKeyChange is to be set, the keyChange value must be derived from the old and new passwords and the engineID of the agent at which the key will be used. The pwchange command is used to generate the keyChange values.

Format



Parameters

-e This flag indicates that the agent for which the keychange value is being defined is identified by engineID rather than by IP address or host name.

-d *n*
This flag indicates what level of debug information is required. Debug tracing is either on or off: 1 causes debug tracing to be generated to the screen of the command issuer (sysout). Debug tracing is off (0) by default.

-p *protocol*
This flag indicates the protocols for which the keychange values are generated. Valid values for *protocol* are:

HMAC-MD5
Generates keychange values for use with the HMAC-MD5 authentication protocol. This is the default.

HMAC-SHA
Generates keychange values for use with the HMAC-SHA authentication protocol.

all Generates both HMAC-MD5 and HMAC-SHA keychange values.

The default is that keychange values for the HMAC-MD5 protocol are generated.

-u *key_usage*
This flag indicates the usage intended for the keychange value. Valid values are:

auth An authentication keychange value

priv A privacy keychange value

all Both authentication and privacy keychange values

Note: There is no difference between a keychange value generated for authentication and a keychange value generated for privacy. However, the length of privacy keychange values depends on whether the keychange value is localized.

-s This flag indicates that output is displayed with additional spaces to improve readability. By default, data is displayed in a condensed format to facilitate cut-and-paste operations on the keychange values onto command lines in shell scripts.

old_password
Specifies the password that was used in generating the key originally. The *password* must be between eight and 255 characters long.

new_password
Specifies the password that will be used in generating the new key. The *password* must be between eight and 255 characters long.

IPaddress
Specifies the IP address in IPv4 dotted decimal or IPv6 colon hexadecimal notation of the agent at the destination host at which the key is to be used.

hostname
Specifies the destination host at which the key is to be used.

engineID
Specifies the engine ID (1–32 octets, 2–64 hexadecimal digits) of the destination

host at which the key is to be used. The engine ID must be a string of 1–32 octets (2–64 hexadecimal digits). If the engine ID is specified, the -e option must also be specified. The default is that the agent identification is not an engine ID.

Usage

The pwchange command generates different output, depending on which protocol and what key usage is selected. Keychange values are typically twice as long as the key to be changed.

Examples

Sample pwchange output:

```
# pwchange oldpassword newpassword 9.67.113.79
Dump of 32 byte HMAC-MD5 authKey keyChange value:
  3eca6ff34b59010d262845210a401656
  78dd9646e31e9f890480a233dbe1114d
```

The value to be set should be passed as a hex value:

```
snmp set usmUserAuthKeyChange.12.0.0.0.2.0.0.0.0.9.67.113.79.2.117.49
\'3eca6ff34b59010d262845210a40165678dd9646e31e9f890480a233dbe1114d\'h
```

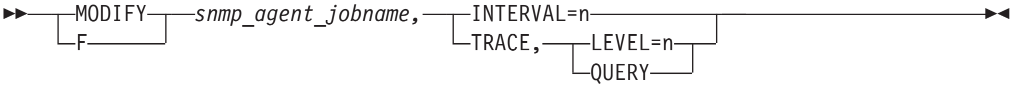
Note: The backslash in the preceding example is required before the single quotation mark to enable z/OS UNIX to correctly interpret the hexadecimal value. (The index of the usmUserTable is made up of the engineID and the ASCII representation of the user name; in this case it is 2 characters long and translates to 117.49.)

Note: pwchange incorporates a random component in generating keys and keyChange values. The output from multiple commands with the same input does not produce duplicate results.

Modifying SNMP agent parameters

Some SNMP agent initialization parameters can be modified while the agent is executing using the MVS MODIFY command. The MODIFY command can also be used to display the current level of SNMP agent tracing.

Format



```
►►──┬─MODIFY─┬──snmp_agent_jobname,──┬─INTERVAL=n───────────┬──────────────────►◄
    └─F──────┘                       └─TRACE,──┬─LEVEL=n─┬──┘
                                               └─QUERY───┘
```

Parameters

snmp_agent_jobname
The SNMP agent being used.

INTERVAL
Specifies an integer in the range 0–10 that indicates the maximum number of minutes before committed configuration changes to the SNMPD.CONF file will

be written out. A value of 0 means that the changes will be written out at the time the SNMP SET request is committed.

TRACE
Indicates SNMP agent tracing is to be queried or changed.

LEVEL
Specifies an integer in the range 0–255 that indicates the level of agent tracing. This corresponds to the -d parameter at agent initialization. See the z/OS Communications Server: IP Configuration Reference for additional guidance on setting the trace level.

QUERY
Requests that the current level of SNMP agent tracing be displayed.

SNMP agent: Management data supported

The following sections describe the type of management data supported by the z/OS Communications Server SNMP agent and subagents and how this data can be used to support network management. The SNMP agent supports objects related to the agent's configuration and the subagents connected to it. The subagents shipped with z/OS Communications Server are:

- The TCP/IP subagent
- The OMPROUTE subagent
- The Network SLAPM2 subagent
- The TN3270 Telnet subagent

The agent and subagents support many MIB objects defined as standard objects in RFCs. Additionally, the SNMP agent and the TCP/IP subagent support nonstandard MIB objects, called Enterprise-specific objects. The complete list of MIB objects supported by the SNMP agent and subagents is in Appendix B, “Management Information Base (MIB) objects,” on page 1025. Additionally, subagents other than those shipped with z/OS Communications Server can communicate with the z/OS Communications Server SNMP agent to extend the MIB objects supported. These subagents must use the Distributed Protocol Interface, as documented in the z/OS Communications Server: IP Programmer's Guide and Reference.

SNMP MIB support

The z/OS Communications Server SNMP agent and subagents support for nonstandard MIB variables is defined in several files shipped with the product. These files are installed into the z/OS UNIX file system in the /usr/lpp/tcpip/samples directory:

- mvstcpip.caps

  This file is the z/OS Communications Server SNMP Capability Statement. It contains the formal SMIv2 definition of the MIBs supported by the SNMP agent and subagents shipped with z/OS Communications Server.
- mvstcpip.mi2

  Contains the formal SMIv2 syntax of the IBM MVS Enterprise-specific MIB extension. This is supported by the TCP/IP subagent.
- mvstcpip.mib

  Contains the formal SMIv1 syntax of the IBM MVS Enterprise-specific MIB extension. This is supported by the TCP/IP subagent.
- mvstn3270.mi2

Contains the SMIv2 syntax for the IBM MVS Enterprise-specific TN3270 MIB. This is supported by the SNMP TN3270 Telnet subagent.

- saMIB.mib

  Contains the formal SMIv1 syntax for the subagent MIB (saMIB) objects. This is supported by the SNMP agent.
- saMIB.mi2

  Contains the formal SMIv2 syntax for the subagent MIB (saMIB) objects. This is supported by the SNMP agent.
- slapm2.mi2

  Contains the formal SMIv2 syntax for NETWORK-SLAPM2-MIB objects. This is supported by the Nework SLAPM2 subagent (nslapm2).
- rfc1592b.mib

  Contains the SMIv1 syntax for the additional information that expands the implementation of RFC 1592 (*Simple Network Management Protocol Distributed Protocol Interface Version 2.0*)in z/OS Communications Server. This is supported by the SNMP agent.
- rfc1592b.mi2

  Contains the SMIv2 syntax for the additional information that expands the implementation of RFC 1592 (*Simple Network Management Protocol Distributed Protocol Interface Version 2.0*) in z/OS Communications Server. This is supported by the SNMP agent.
- ibm3172.mi2

  Contains the SMIv2 syntax for the 3172 Enterprise-specific MIB objects. This is supported by the SNMP agent.
- ibm3172.mib

  Contains the SMIv1 syntax for the 3172 Enterprise-specific MIB objects. This is supported by the SNMP agent.

TCP/IP subagent

The TCP/IP subagent supports SNMP management data from both standard and Enterprise-specific SNMP Management Information Base (MIB) modules. The data defined in MIB modules are called MIB objects. The standard MIB modules are those published by the IETF in RFCs. Some of the Enterprise-specific MIB objects extend the standard MIBs by providing additional management information. Other Enterprise-specific MIB objects provide management information specific to the z/OS Communications Server TCP/IP stack implementation, such as:

- The ability to perform a remote ping request to provide response time data between two remote hosts.
- Support for TCP/IP stack configuration parameters. The Enterprise-specific MIB defines several MIB objects that correspond to parameters on Profile configuration statements such as IPCONFIG, IPCONFIG6, TCPCONFIG, UDPCONFIG and so on. For some of these MIB objects, an **snmp** set command can be issued to remotely change the configured value.
- Retrieval of IBM 3172 Interconnect Controller data.
- Retrieval of OSA data.
- Retrieval of dynamic VIPA and sysplex distributor data.

Details of the subagent support for both the standard and Enterprise-specific MIB data can be found in the SNMP Agent Capabilities statement, which is installed in the z/OS UNIX file system as file /usr/lpp/tcpip/samples/mvstcpip.caps. Most of the Enterprise-specific MIB data mentioned in this section is defined in the IBM MVS TCP/IP Enterprise-specific MIB module. This MIB module is installed in the z/OS UNIX file system directory /usr/lpp/tcpip/samples as file mvstcpip.mi2.

The TCP/IP subagent supports only the following types of IP addresses from the INET-ADDRESS-MIB (RFC 4001):

- unknown - Normally used for local or remote IP addresses of TCP Listeners and UDP endpoints, where the socket has not been bound to a local IP address.
- ipv4 - IPv4 addresses
- ipv6 - IPv6 addresses, except for link-local
- ipv6z - IPv6 link-local addresses, where the zone index value is the SNMP interface index of the associated interface.

TCP/IP subagent: Management data supported

The following items are the main areas of the TCP/IP stack for which MIB data is supported:

- IP/ICMP/Route MIB data

  The subagent supports IP/ICMP MIB data from the IP-MIB in RFC 4293, and some additional IP counters from the Enterprise-specific MIB. The subagent supports Route MIB data from the IP-FORWARD-MIB in RFC 4292, and from the TCP/IP Enterprise-specific MIB.
- Interface MIB data

  The subagent supports interface (IF) MIB data from the IF-MIB in RFC 2233. The TCP/IP Enterprise-specific MIB defines the following additional MIB data:
  - Information from the DEVICE, LINK, and INTERFACE profile statements.
  - Multicast group information per interface.
  - Packet trace parameters per interface
  - ibmMvsTcpipIntfUp and ibmMvsTcpipIntfDown notifications, which include the name of the interface whose state is changing.

  The TCP/IP stack assigns the interface index values that are used to identify each interface for this data as each interface is defined. The values might not be continuous. For example, the TCP/IP stack reserves some interface index values for potential interfaces that are used with the Shared Memory Communications over Remote Direct Memory Access (SMC-R) function.

  Although the subagent supports interface state change notifications, these notifications are not created for VIPA interfaces (static or dynamic). This is because VIPA interfaces do not change operational state. If they are successfully defined, then they are always active and they cannot be stopped by using the VARY TCPIP,,STOP command. This affects the following interface state change notifications:
  - The linkup and linkdown notifications from the IF-MIB
  - The ibmMvsTcpipIntfUp and ibmMvsTcpipIntfDown notifications from the TCP/IP Enterprise-specific MIB

  Proprietary status change notifications are still created for dynamic VIPA interfaces. See “SNMP Enterprise-specific trap types” on page 1066 for more information about these notifications.
- TCP MIB data

  The subagent supports the TCP MIB data from the TCP-MIB in RFC 4022. The TCP global counters in the TCP-MIB reflect both IPv4 and IPv6 processing. The

Enterprise-specific MIB augments the standard IPv4-only and version-neutral TCP connection table; provides a TCP Listener table with server MIB data; and provides additional TCP stack counters.

- UDP MIB data

  The subagent supports the UDP MIB data from the UDP-MIB in RFC 4113. The UDP global counters in the UDP-MIB reflect both IPv4 and IPv6 processing. The Enterprise-specific MIB augments the standard UDP listener table (IPv4-only) and the version-neutral UDP endpoint table, and also provides multicast information.

- TCP/IP stack configuration data

  The TCP/IP Enterprise-specific MIB defines MIB objects that support the following configuration data:

  - Data from Profile configuration statements such as IPCONFIG, IPCONFIG6, SACONFIG, TCPCONFIG, and UDPCONFIG
  - TCP/IP stack name
  - MVS image name
  - XCF group name used by the stack when joining the sysplex

SNMP remote PING

SNMP remote PING is a function of the TCP/IP subagent that gives an SNMP manager the ability to obtain the round-trip response time for an ICMP echo request message (PING) from an SNMP agent to a destination IP address.

The SNMP remote PING function is a valuable tool in an Enterprise network that provides centralized management services because it gives a third-party (SNMP manager) system the ability to request that a PING operation be performed on a remote system running z/OS. The remote system must be running the SNMP agent and the TCP/IP subagent.

For example, if there are three hosts (A, B, and C) as shown in Figure 5 on page 983, you can obtain the response time between the two remote hosts. In this example, your host is running the SNMP manager function (Host A), Host B is running the SNMP agent and TCP/IP subagent functions, and Host C is some arbitrary remote host. The standard PING function enables Host A to obtain the round-trip response time from A to B and from A to C, but not from B to C. With the SNMP remote PING function on the TCP/IP subagent, Host A can obtain the round-trip response time from B to C.

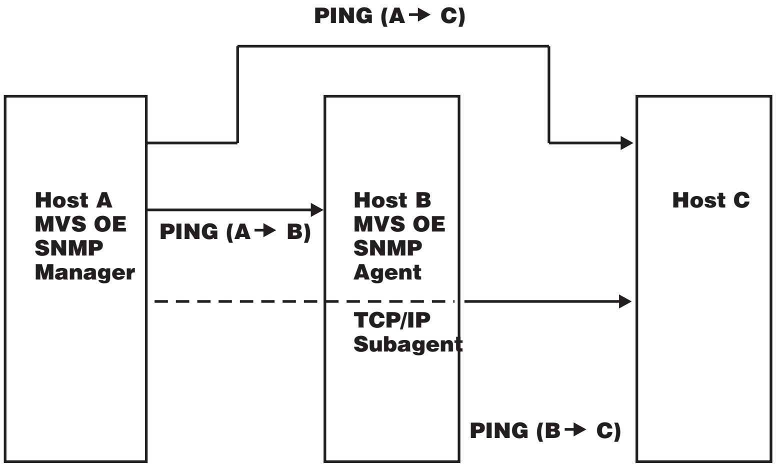


Figure 5. SNMP remote PING function

With the SNMP remote PING function, you can specify the size of the packet, in bytes, that is sent in the ICMP echo request message and the time period, in seconds, to wait for that ICMP echo request message to return from the requested destination address.

SNMP remote PING format

To send a remote **ping** command, use the NetView SNMP GET command or the z/OS UNIX **snmp get** command. Specify *ibmMvsRemPingResponseTime* as the mib_variable on the command. The earlier *ibmMvsRPingResponseTime* MIB object can also be specified on the command but this MIB object supports only IPv4 ping requests and has been deprecated. Both MIB objects are defined in the IBM MVS TCP/IP Enterprise-specific MIB module. The object identifier (OID) of the *ibmMvsRemPingResponseTime* MIB object in ASN.1 notation is 1.3.6.1.4.1.2.6.19.2.2.1.2.1.5.

```
            ┌─-d 0───────────┐  ┌─-h localhost─┐  ┌─-r 2───────────┐
►►──osnmp───┼────────────────┼──┼──────────────┼──┼────────────────┼──────────────────►
            └─-d debug_level─┘  └─-h host name─┘  └─-r retry number┘

                                                          ┌──────────────────┐
   ┌─-c public─────────┐  ┌─-t 3────────────┐             ▼                  │
►──┼───────────────────┼──┼─────────────────┼──┬────┬──get───mib_variable───┴──────►◄
   └─-c community_name─┘  └─-t timeout value┘  └─-v─┘
```

SNMP remote PING parameters

mib_variable
: Specifies one or more MIB variable names to be retrieved. You can specify the names in textual form or ASN.1 notation.

For the remote ping object, a three-part index is required, with each part separated by periods (.), as in the following example:

```
snmp -h host_name get ibmMvsRemPingResponseTime.packet_size.time_out.ip_address
```

Note: To find a description of the other parameters, see "Parameters" on page 954.

The following list describes the get portion of the command, including the three-part index for the remote ping object:

Instance	Description
ibmMvsRemPingResponseTime	Specifies that the remote ping command should be issued.
packet_size	Specifies the packet size of the ping request.
time_out	Specifies the timeout value, in seconds, for the ping request.
ip_address	Specifies the IP address of the remote host to which the ping request is directed. The IP address is comprised of the following three parts: 1. IP address type from the INET-ADDRESS-MIB. The currently supported types are: 1 - ipv4, 2 - ipv6, 4 - ipv6z (link-local). 2. IP address length: 4 - ipv4, 16 - ipv6, 20 - ipv6z. 3. IP address, where each octet of the address is converted to decimal and separated from the other octets by a period.

SNMP remote PING example

The example shows how to use the z/OS UNIX **snmp get** command to perform a remote ping to an IPv4 remote host:

```
snmp -h mvs1 -c mvs150 get ibmMvsRemPingResponseTime.2048.5.1.4.9.37.33.175
```

where:

```
host_name = mvs1

community_name  = mvs150

mib_variable  = ibmMvsRemPingResponseTime.2048.5.1.4.9.37.33.175  where:

         packet size     = 2048 bytes
         time-out        = 5 seconds
         ip_address      = 1. (IP address type is ipv4)
                           4. (IP address length is 4 for ipv4)
                           9.37.33.175 (IPv4 address)
```

The expected response is as follows:

```
1.3.6.1.4.1.2.6.19.2.2.1.2.1.5.2048.5.1.4.9.37.33.175=33
```

The variable value in the previous example is a positive value (33) indicating a successful response. The variable number, when positive, is the round-trip response time, in milliseconds, from the SNMP agent host system to the requested destination IP address. The following example shows how to use the z/OS UNIX **snmp get** command to perform a remote ping to IPv6 remote host 2001:0DB8::1 :

```
snmp -h mvs1 -c mvs150 get ibmMvsRemPingResponseTime.2048.5.2.16.32.1.13.184.0.0.0.0.0.0.0.0.0.0.0.1
```

where:

```
host_name = mvs1
community_name  = mvs150
mib_variable  = ibmMvsRemPingResponseTime.2048.5.2.16.32.1.13.184.0.0.0.0.0.0.0.0.0.0.0.1 where:
                   packet size     = 2048 bytes
                   time-out        = 5 seconds
```

```
ip_address      = 2. (IP address type is ipv6)
                 16. (IP address length)
                    32.1.13.184.0.0.0.0.0.0.0.0.0.0.0.1 (IP address)
```

The expected response is as follows:

```
1.3.6.1.4.1.2.6.19.2.2.1.2.1.5.2048.5.2.16.254.192.0.0.0.0.0.0.0.0.0.0.0.0.0.1 = 33
```

The variable value can be a negative integer indicating that a failure has occurred. A negative integer is a result of the SNMP agent or TCP/IP subagent detecting either an internal error, an incorrect MIB instance format, an ICMP echo request timeout, an incorrect packet size value, an incorrect timeout value or an incorrect destination IP address. See Table 20 for a description of what the variable value can represent.

Table 20. SNMP Get command responses for variable value

Returned value	Description	Condition	Valid input
>0 (milliseconds)	Round-Trip Response Time	Success	N/A
-1	Internal error	Failure	N/A
-2	ICMP echo request timed out	Failure	N/A
-3	Destination was IPv6 but subagent stack not IPv6 enabled	Failure	N/A
-4	Incorrect packet size	Failure	0, 16–4096 (bytes)
-5	Incorrect timeout	Failure	0, 3–15 (seconds)
-6	Unknown destination IP address	Failure	IP address types of 1, 2, or 4; IP address lengths of 4, 16, or 20; fully-qualified IP address.
-7	Incorrect MIB instance format	Failure	Packet size.timeout.IP address type.length.address

Note: The packet size and the timeout in the *mib_variable* value part of the **snmp get** command can have a value of 0, which indicates that the default values are 256 bytes and 10 seconds, respectively.

Interface layering

In the SNMP framework, the most fundamental MIB table is the interfaces table. The TCP/IP subagent supports interface MIB data from the IF-MIB from RFC 2233. For more information, see Appendix A, “SNMP capability statement,” on page 1005 for a list of supported IF-MIB objects. RFC 2233 provides the following basic interface tables:

- The ifTable and ifXTable
- The ifStackTable, which shows how interfaces are layered

The TCP/IP subagent interface layering implementation is explained by the following example, in which DEVICE, LINK, and INTERFACE profile statements are specified in the TCP/IP profile data set. This example defines only OSA-Express QDIO Ethernet interfaces.

```
DEVICE OSA4C   MPCIPA
LINK LV4OSA4C  IPAQENET  OSA4C
INTERFACE LV6OSA4C DEFINE IPAQENET6 PORTNAME OSA4C
     IPADDR 2001:0DB8:0:1:0009:0067:0115:0066

INTERFACE LV4OSA8D DEFINE IPAQENET  PORTNAME OSA8D
     IPADDR 9.67.116.66/24
INTERFACE LV6OSA8D DEFINE IPAQENET6 PORTNAME OSA8D
     IPADDR 2001:0DB8:0:1:0009:0067:0116:0066
```

Figure 6. TCP/IP subagent interface layering implementation example

The previous example code would create the interface entries described in Table 21:

Table 21. ifType interface entries

ifIndex	ifType	Description
1	53 (propVirtual)	LOOPBACK device
2	24 (softwareLOOPBACK)	LOOPBACK link
3	24 (softwareLOOPBACK)	LOOPBACK6 interface
4	53 (propVirtual)	MPCIPA device OSA4C
5	6 (ethernetCsmacd)	IPAQENET link LV4OSA4C
6	6 (ethernetCsmacd)	IPAQENET6 interface LV6OSA4C
7	53 (propVirtual)	OSA-Express port interface OSA8D
8	6 (ethernetCsmacd)	IPAQENET interface LV4OSA8D
9	6 (ethernetCsmacd)	IPAQENET6 interface LV6OSA8D

The ifType values indicate the interface type and are assigned by the Internet Assigned Numbers Authority (IANA). In z/OS Communications Server, a DEVICE profile statement has a corresponding entry in the IF-MIB interface tables. The lower-layer interfaces for the DEVICE (defined by LINK statements) are also defined as interface entries, stacked below the device entry.

For INTERFACE profile statements (other than IPAQENET or IPAQENET6), only an interface entry is created in the interface tables. There is no higher-layer device entry associated with the interface

For OSA-Express QDIO Ethernet interfaces (IPAQENET and IPAQENET6), the associated OSA-Express port is always the highest-layer entry in the interface tables. The OSA-Express port entry is either created as the result of a DEVICE profile statement or is dynamically created when only INTERFACE profile statements are used to define IPAQENET or IPAQENET6 interfaces. For dynamically created OSA-Express port entries, the ifName MIB object is set to the PORTNAME parameter value from the INTERFACE profile statements. All of the interfaces for the same OSA-Express port are stacked below the same OSA-Express port entry.

The ifTable and ifXTable counters for each device or OSA-Express port entry reflect the sum of the counters for the underlying links or interfaces. See the IF-MIB table for a detailed explanation of how the ifStackTable is used to display interface relationships.

Table 21 shows that a LOOPBACK device entry and link entry were created in the IF-MIB tables, even though the interfaces were not explicitly defined. TCP/IP automatically generated these entries. Because the TCP/IP stack was enabled for IPv6 support, a LOOPBACK6 interface entry was also automatically generated.

When an ATM DEVICE is defined, two subordinate interface entries are created below it, AAL5 and ATM. AAL5 and ATM are UNI-defined layers that exist physically in an ATM port. The ifEntry and ifXEntry counters reflect traffic though the port. If the ATM DEVICE is configured for LAN emulation mode, two additional subordinate layers might be created after the AAL5 and ATM layers. These additional layers represent emulated link interfaces. The counter data for all of these ATM subordinate layers is obtained directly from the Open Systems Adapter/Support Facility (OSA/SF). See "ATM-specific management data" on page 989 for more information.

IBM 3172 Enterprise-specific MIB variables

The IBM 3172 interconnect controller maintains a set of Enterprise-specific MIB variables. The SNMP agent can act as a proxy agent to retrieve these variables from the 3172 device. You can issue either a GET or GETNext command to retrieve the 3172 variables. The 3172 variable names can be included in a GET or GETNext command that also contains standard MIB variable names. See Appendix B, "Management Information Base (MIB) objects," on page 1025 for a description of the 3172 Enterprise-specific MIB variables.

The 3172 variables are referenced by a single element instance identifier, for example, (.1, .2, .3). This identifier is the interface index, ifIndex, assigned to the LAN channel system (LCS) device and links by TCP/IP. TCP/IP assigns ifIndex values to its devices and links based on the order in which they are defined to TCP/IP. The following example shows the profile statements and the ifIndex values that would be assigned:

```
                                                 ifIndex
                                                 -------
DEVICE LCS1     LCS          120     NETMAN         3
DEVICE CTCD00   CTC D00                             4
LINK CTC1       CTC 1 CTCD00 IFSPEED   12345        5
LINK TR1  IBMTR      1 LCS1                         6
```

For objects which pertain to the entire 3172, the instance identifier is the ifIndex of the LCS device. In the example above, this is an ifIndex value of 3.

For counter objects related to a specific link interface, the instance identifier is the ifIndex value of that link. In the example above, this is an ifIndex value of 6.

If a GET command is issued for a counter object using an instance identifier of a link that does not support the 3172 objects, a response of NO SUCH INSTANCE is returned from the SNMP agent.

If a GETNext command is issued, the links that do not support the 3172 objects are skipped and the NEXT link that does support the 3172 objects is returned.

If an error occurs accessing a 3172 variable from the 3172 (either an error return code is received from the 3172 device or no response is received from the 3172 device), an error code of GEN ERROR is returned to the client in the SNMP response PDU for that variable. An error message containing more specific information about the error that occurred is written to the syslog daemon if SNMP subagent tracing has been activated by the ITRACE profile statement. Several of the potential error conditions reference the 3172 MIB variable by the 3172 attribute index. See Appendix C, “IBM 3172 attribute index,” on page 1063 for a list of the 3172 attribute indices and the corresponding MIB variable names.

OSA feature management data

The TCP/IP subagent supports management data for following types of OSA features:

- OSA-2 ATM
- OSA-Express Gigabit
- OSA-Express fast Ethernet (QDIO and non-QDIO modes)
- OSA-Express ATM (LAN emulation mode only)
- OSA-Express2 or later Gigabit

The TCP/IP subagent requires the OSA/SF product to retrieve the management data from the OSA features. The OSA product also provides an SNMP subagent, the OSA-Express Direct subagent, that supports management data for OSA-Express features and OSA-Express2 or later features. The MVS-started procedure name of this subagent is IOBSNMP. You should use the OSA-Express Direct subagent to obtain OSA management data, because it communicates directly with the OSA features and does not require the OSA/SF and IOASNMP applications. If you are using the TCP/IP subagent's OSA management data support and decide to switch to the OSA-Express Direct subagent, you no longer need to start the OSA/SF address space or the OSA IOASNMP application. For a complete understanding of the management data provided by the OSA-Express Direct subagent, see the zEnterprise System and System z10 OSA-Express Customer's Guide and Reference.

For the TCP/IP subagent OSA adapter support, some of the management data is defined in standard RFCs and the remaining data is defined in the IBM MVS TCP/IP Enterprise-specific MIB. See “SNMP MIB support” on page 979 for information on locating the IBM MVS TCP/IP Enterprise-specific MIB. Some of the management data values are provided by TCP/IP and some by OSA/SF.

See Step 4: Configure the Open Systems Adapter (OSA) support in the z/OS Communications Server: IP Configuration Guide for information on configuring the SNMP subagent to communicate with OSA/SF. See Appendix F, “Related protocol specifications,” on page 1073 for information about RFCs.

The following MIB tables describe the supported OSA feature management data. The osaexpChannelTable, osaexpPerfTable, and osaexpEthPortTable, which are defined in the IBM MVS Enterprise-specific MIB, have been deprecated because the same data is supported by the OSA-Express Direct subagent. The MIB data supported by the OSA-Express Direct subagent is defined in the OSA Enterprise-specific MIB module, IBM-OSA-MIB. See the zEnterprise System and System z10 OSA-Express Customer's Guide and Reference for instructions about obtaining a copy of the IBM-OSA-MIB module.

- **osaexpChannelTable**

  An entry in this table is created for every OSA-Express Ethernet or ATM port that is in use by the TCP/IP stack. The table contains descriptive and performance data. The values are retrieved from OSA/SF. This table is indexed by the ifIndex of the device or OSA-Express port interface and is defined in the IBM MVS Enterprise-specific MIB.
- **osaexpPerfTable**

  An entry in this table is created for every OSA-Express Ethernet or ATM port that is in use by the TCP/IP stack, one entry per LPAR to which the adapter is defined. The table contains performance data per LPAR's use of the adapter and the values are retrieved from OSA/SF. This table is indexed by the ifIndex of the device or OSA-Express port interface concatenated with the decimal LPAR number. This table is defined in the IBM MVS Enterprise-specific MIB.

- **osaexpEthPortTable**

  An entry in this table is created for every OSA-Express Ethernet port that is in use by the TCP/IP stack. The table contains descriptive and performance data related to the adapter's physical port and the values are retrieved from OSA/SF. This table is indexed by the ifIndex of the device or OSA-Express port interface and is defined in the IBM MVS Enterprise-specific MIB.
- **osaexpEthSnaTable**

  An entry in this table is created for every OSA-Express Ethernet feature that is configured for SNA and defined to TCP/IP by LCS DEVICE and Ethernet LINK profile statements. The values are retrieved from OSA/SF. This table is indexed by the ifIndex of the device interface and is defined in the IBM MVS Enterprise-specific MIB.
- **Interface Table Data**

  An entry is created in the ifTable and ifXTable tables for the following statements:

  - Every DEVICE and LINK profile statement that represents an OSA-Express feature.
  - Every INTERFACE profile statement that represents an OSA-Express feature. If only INTERFACE profile statements are used to define the interfaces, then an additional OSA-Express port entry is dynamically created in the tables.

  The ifTable and ifXTable data for OSA-Express interfaces that are used by TCP/IP for data transport is retrieved from TCP/IP. Interface Table Data is defined in the IF-MIB from RFC 2233.
- **dot3StatsTable**

  An entry in this Ethernet table is created for every OSA-Express Ethernet port that is in use by the TCP/IP stack. The values are retrieved from OSA/SF. This table is indexed by the ifIndex of the device or OSA-Express port interface and is defined in the EtherLike-MIB (RFC 2665).

  The OSA-Express Direct SNMP subagent can also support the dot3StatsTable if the OSA feature LIC level that you are using supports it. In that case, the OSA-Express Direct SNMP subagent takes over ownership of the dot3StatsTable MIB data. If the OSA-Express Direct subagent is not active, or was active and then terminated, the TCP/IP subagent takes over the ownership of the data. The movement of ownership of this MIB data between the TCP/IP subagent and the OSA-Express Direct subagent should be transparent and SNMP requests for the data continue to be processed. For more information about using the OSA-Express Direct SNMP subagent, see the zEnterprise System and System z10 OSA-Express Customer's Guide and Reference.

ATM-specific management data

Some OSA-Express ATM management data is represented in the osaexpChannelTable and the osaexpPerfTable. Outside of the Interface Table Data, the rest of the OSA-Express ATM data and all of the OSA-2 ATM data are represented in the following tables.

- **osasfChannelTable**

  An entry in this table is created for every OSA-2 ATM DEVICE profile statement. Each ATM DEVICE statement represents one ATM adapter card externally through SNMP. This table is indexed by the ifIndex of the ATM DEVICE and the values are retrieved from OSA/SF. This table is defined in the IBM MVS Enterprise-specific MIB.

- **osasfPvcTable**

  An entry in this table is created for every PVC defined for an OSA-Express or OSA-2 ATM Port. Indexing is by the ifIndex of the AAL5 layer and pvcName. The values are retrieved from OSA/SF. Each port has a limit of 256 PVCs. This table is defined in the IBM MVS Enterprise-specific MIB.
- **osasfPortTable**

  An entry in this table is created for every OSA-Express or OSA-2 ATM DEVICE interface. Indexing is by the ifIndex of the AAL5 interface layer. This table is defined in the IBM MVS Enterprise-specific MIB.
- **ibmMvsAtmSnaLeTable**

  One entry in this table is created for every OSA-Express or OSA-2 ATM LAN Emulation interface where the ATM port is configured for SNA and LAN Emulation mode. Indexing is by the ifIndex of the ATM LAN Emulation interface. This table is defined in the IBM MVS Enterprise-specific MIB.
- **ibmMvsAtmLecConfigTable**

  One entry in this table is created for every OSA-Express or OSA-2 ATM LAN Emulation interface, where the ATM port is configured for LAN Emulation mode. This table is modeled after the LEC Config Table from the LAN Emulation MIB defined by the ATM Forum. Indexing is by the ifIndex of the ATM LAN Emulation interface. This table is defined in the IBM MVS Enterprise-specific MIB.
- **ibmMvsAtmLecStatusTable**

  One entry in this table is created for every OSA-Express or OSA-2 ATM LAN Emulation interface, where the ATM port is configured for LAN Emulation mode. This table is modeled after the LEC Status Table from the LAN Emulation MIB defined by the ATM Forum. Indexing is by the ifIndex of the ATM LAN Emulation interface. This table is defined in the IBM MVS Enterprise-specific MIB.
- **ibmMvsAtmLecStatisticsTable**

  One entry in this table is created for every OSA-Express or OSA-2 ATM LAN Emulation interface, where the ATM port is configured for LAN Emulation mode. This table is modeled after the LEC Statistics Table from the LAN Emulation MIB defined by the ATM Forum. Indexing is by the ifIndex of the ATM LAN Emulation interface. This table is defined in the IBM MVS Enterprise-specific MIB.
- **ibmMvsAtmLecServerTable**

  One entry in this table is created for every OSA-Express or OSA-2 ATM LAN emulation interface, where the ATM port is configured for LAN emulation mode. This table is modeled after the LEC server table from the LAN emulation MIB defined by the ATM Forum. Indexing is by the ifIndex of the ATM LAN emulation interface. This table is defined in the IBM MVS Enterprise-specific MIB.
- **ibmMvsAtmLecMacAddressTable**

  One entry in this table is created for every OSA-Express or OSA-2 ATM LAN emulation interface, where the ATM port is configured for LAN emulation mode. This table is modeled after the LEC Mac Address Table from the LAN emulation MIB defined by the ATM forum. Indexing is by the ifIndex of the ATM LAN emulation interface. This table is defined in the IBM MVS Enterprise-specific MIB.

- **Interface Table Data**

  ifTable and ifXTable data are retrieved from OSA/SF for the AAL5, ATM, and LAN emulation interfaces subordinate to an ATM DEVICE interface. ifTable and ifXTable data for ATM DEVICE and LINK interfaces used by TCP/IP for data transport is retrieved from TCP/IP. Interface table data is from the ifMIB - RFC 2233.
- **atmInterfaceConfTable**

  One entry in this table is created for every ATM LINK interface. It is, however, indexed by the ifIndex of the AAL5 interface entry. This table is defined in the atmMIB - RFC 1695 (*Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2*).
- **ipoaLisTable**

  An entry in this table is created for every ATMLIS statement whose LIS name is referenced on an ATM LINK statement. The ipoaLisTable is from the ipoaMIB - RFC 2320 (*Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2*).
- **ipoaLisIfMappingTable**

  An entry in this table is created for every ATM LINK statement, which includes an LIS name. The ipoaLisIfMappingTable is from the ipoaMIB - RFC 2320 (*Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2*).
- **ipoaArpClientTable**

  An entry in this table is created for every local IP address that is assigned to an ATM interface (for every LINK ATM statement on a DEVICE ATM). The ipoaArpClientTable is from the ipoaMIB - RFC 2320 (*Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2*).
- **ipoaArpRemoteServerTable**

  An entry in this table is created for every TCP/IP link to an ATMARP remote server. The ipoaArpRemoteServerTable is from the ipoaMIB - RFC 2320 (*Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2*).
- **ipoaVcTable**

  An entry in this table is created for each ATM VC connection. The ipoaVcTable is from the ipoaMIB - RFC 2320 (*Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2*).
- **ipoaConfigPvcTable**

  An entry in this table is created for each ATM VC connection, which is a permanent VC. The ipoaConfigPvcTable is from the ipoaMIB - RFC 2320 (*Definitions of Managed Objects for Classical IP and ARP over ATM Using SMIv2*).

ATM port IP address assignment:
SNMP provides a method for assigning an IP address to an OSA-2 ATM port. The ATM port reports the IP address, atmfMyIpNmAddress, as specified by the ATM Forum User-Network Interface (UNI) specification. UNI defines an Interim Local Management Interface (ILMI) layer that provides an MIB that can be accessed directly over an ATM network by way of an SNMP request.

To specify an IP address for an ATM port, use the **snmp set** command against the ibmMvsAtmOsasfPortIpAddress MIB object (this MIB object is defined in the IBM MVS Enterprise-specific MIB). Once an IP address is set, the ATM port remembers the IP address and it does not have to be reset. Make sure you issue the **snmp set** command on the MVS image where the managing OSA/SF for the ATM device is running. For information about the **snmp set** command, see “The z/OS UNIX snmp command” on page 953.

ATM trap notification from OSA/SF:
Asynchronous events are forwarded from OSA/SF to the SNMP TCP/IP subagents. These events are converted to traps and sent to the snmp agent associated with the TCP/IP instance receiving the notification, for forwarding. The traps supported for ATM Management are:

- **Permanent Virtual Circuit (PVC) creation--ibmMvsAtmOsasfAtmPvcCreate Trap**

  This trap is supported only for ATM OSA-2 adapters. It is not supported for OSA-Express ATM155 adapters. An ibmMvsAtmOsasfAtmPvcCreate notification is generated when OSA/SF sends an asynchronous notification to a subagent that a PVC was created for a given OSA-2 ATM Port.
- **Permanent Virtual Circuit (PVC) deletion--ibmMvsAtmOsasfAtmPvcDelete Trap**

  This trap is supported only for ATM OSA-2 adapters. It is not supported for OSA-Express ATM155 adapters. An ibmMvsAtmOsasfAtmPvcDelete notification is generated when a PVC is deleted.

Note: The TCP/IP subagent discards any notification received for an ATM port that is not properly defined through an ATM DEVICE statement.

Dynamic VIPA and sysplex distributor management data

The TCP/IP subagent supports dynamic VIPA (DVIPA) and sysplex distributor management data from the IBM MVS TCP/IP Enterprise-specific MIB. See Appendix B, “Management Information Base (MIB) objects,” on page 1025 for a list of all the DVIPA MIB objects. See “SNMP Enterprise-specific trap types” on page 1066 for a description of all the supported traps. The following describe new MIB tables:

`ibmMvsDVIPATable`
An entry is created in this table for each dynamic VIPA defined to a TCP/IP stack.

`ibmMvsDVIPARouteConfTable`
An entry is created in this table for every VIPAROUTE profile statement.

`ibmMvsDVIPARangeConfTable`
An entry is created in this table for every IPv4 dynamic VIPA address range defined by the VIPARANGE profile statement. This table cannot support IPv6 entries as it uses an address mask as part of the index value, and address masks do not apply to IPv6. Because of this, support for this table has been deprecated. This means the data in the table will continue to be supported but management applications should not implement new support for this table. Instead management applications should support the ibmMvsDVIPARangeConfigTable.

`ibmMvsDVIPARangeConfigTable`
An entry is created in this table for every IPv4 and IPv6 dynamic VIPA address range defined by the VIPARANGE profile statement.

`ibmMvsDVIPADistConfTable`
An entry is created in this table for every dynamic VIPA and port for which connection requests are to be distributed to other TCP/IP stacks as defined by a VIPADISTRIBUTE profile statement.

`ibmMvsDVIPAConnRoutingTable`
Each entry in this table represents a dynamic VIPA TCP connection. Entries will be added to the table only for dynamic VIPA connections for which

MOVEABLE IMMEDIATE or NONDISRUPTIVE was specific in the TCP/IP profile. On a sysplex distributor routing stack, there is an entry in this table for every connection being routed through the distributor. On a stack taking over a dynamic VIPA, there is an entry in this table for every connection to the dynamic VIPA. On a sysplex distributor target stack or a stack that is in the process of giving up a dynamic VIPA, there is an entry in this table for every connection for which the stack is an endpoint.

ibmMvsDVIPADistPortTable
An entry is created in this table for every target stack per distributed dynamic VIPA IP address and port. This table is supported only by stacks that are distributing connection requests as part of the sysplex distributor function. This table is not supported by stacks that are targets only of the sysplex distributor function.

There are also scalar MIB objects to support the sysplex distributor Service Manager function and to control generation of dynamic VIPA traps.

OMPROUTE subagent

The OMPROUTE subagent provides an alternative to DISPLAY commands for displaying Open Shortest Path First (OSPF) protocol configuration and state information. The subagent implements the Management Information Base (MIB) variables defined in RFC 1850 (*OSPF Version 2 Management Information Base*).

Network SLAPM2 subagent

The Network SLAPM2 subagent provides support for the Network Service Level Agreement Performance Monitor MIB (NETWORK-SLAPM2-MIB).

This MIB provides information about defined policy rules, and performance statistics for TCP and UDP connections that map to active policies. It can monitor various types of policy rules for TCP connections. When monitoring entry is created, a set of gauges and counters related to the policy rule being monitored are maintained. The monitor table entries can be configured to send *not ok* SNMP traps when a specified threshold related to the gauges goes above its high threshold, and then an *ok* trap is sent when it goes below its low threshold. SNMP traps can be configured when a policy rule monitored entry is deleted or a policy rule static entry is deleted. See the Network SLAPM2 subagent section of the z/OS Communications Server: IP Configuration Guide for more information about the Network SLAPM2-MIB subagent.

TN3270 Telnet subagent

The TN3270 Telnet subagent provides support for the TN3270 Server transaction management data defined in the IBM MVS Enterprise-specific TN3270 MIB. The IBM MVS TN3270 Enterprise-specific MIB is installed in the z/OS UNIX file system as file /usr/lpp/tcpip/samples/mvstn3270.mi2. See Appendix B, “Management Information Base (MIB) objects,” on page 1025 for a list of all the TN3270 Server SNMP MIB objects defined in this MIB.

The following describe the new MIB tables:

ibmMvsTN3270ConnTable
An entry is created in this table for each TN3270 connection being monitored. Each entry contains transaction data for a specific connection.

ibmMvsTN3270MonGroupTable
An entry is created in this table for every Monitor Group defined by a TN3270 Server MONITORGROUP profile statement.

For more details about the data that is defined in the IBM MVS Enterprise-specific TN3270 MIB, and about how to cause connections to be monitored, see the information about accessing remote hosts using Telnet in the z/OS Communications Server: IP Configuration Guide.

The trap forwarder daemon (TRAPFWD)

The trap forwarder daemon receives a trap on a specified port and forwards it to multiple ports on the same host and on different hosts. This allows multiple SNMP managers at one IP address to be able to receive all of the traps sent to one port.

When traps are forwarded, the originating IP address on the forwarded datagram will be that of the trap forwarder daemon, not the originating agent. SNMPv1 format traps are not typically a problem; the trap PDU contains the IP address of the originating agent. However, SNMPv2 format traps do not contain the agent's IP address. For SNMPv2 format traps, the trap forwarder daemon can be configured to append the originating agent's IP address to the datagram that gets forwarded. The receiving management application must have logic to obtain the agent's IP address from the end of the datagram. The default is to pass the datagram that was received without adding anything to it.

For the trap forwarder daemon to forward the datagram with the agent address, the ADD_RECV_FROM_INFO option must be coded on the destination address line in the TRAPFWD.CONF configuration file. See the z/OS Communications Server: IP Configuration Reference for statement syntax. The receiving management application must parse the received datagram, along with the appended agent address. The address field contains the originating agent address, followed by the length of the address. By examining the last four bytes of the received datagram, the management application can determine the length of the agent address.

Chapter 8. SNTP daemon: Simple Network Time Protocol

SNTPD is a TCP/IP daemon that is used to synchronize time between a client and a server. SNTP (Simple Network Time Protocol) is a protocol for synchronizing clocks across a WAN or a LAN through a specific formatted message.

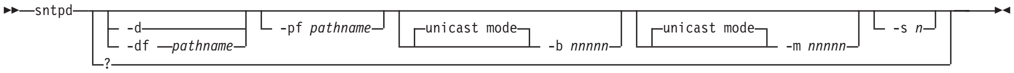
An External Time Reference (ETR) named *stratum 0*, is chosen as the highest timer reference. A *stratum 1* server is a server attached to a *stratum 0* timer. For example, the z/OS sysplex timer could be a *stratum 0* timer and z/OS Communications Server would be a *stratum 1* server. A client attached to *stratum 1* server can also be a *stratum 2* server, and so on. SNTP uses UDP packets for data transfer with the well-known port number 123. RFC 2030 (Mills 1996) describes SNTP. You can start SNTPD from the z/OS UNIX shell or as a started procedure. Each of these methods is described in the z/OS Communications Server: IP Configuration Reference.

The z/OS UNIX sntpd command: Simple Network Time Protocol

The z/OS UNIX **sntpd** command is used to start the sntp daemon.

Note: TCP/IP must be started prior to starting SNTPD.

Format



Parameters

-? Specifies the command help.

-d
Enables debugging. Debug messages go to the syslog daemon.

-df *pathname*
Enables debugging. Debug messages go to the specified file location. For example:

```
-df /var/sntpd.debug
```

-pf *pathname*
z/OS UNIX file system path for the pid file. For example:

```
 -pf /var
```

-b *nnnnn*
Act in broadcast mode. Send local broadcasts on all interfaces every *nnnnn* seconds. Valid values are in the range 1 – 16 284. Listen for requests and respond with unicast replies.

-m *nnnnn*
Act in multicast mode. Send multicast updates (TTL = 1) on all interfaces every *nnnnn* seconds. Valid values are in the range 1 – 16 284. Listen for requests and respond with unicast replies.

-s *n*

Use *n* as the stratum level in all replies sent by the server. Valid values for *n* are in the range 1 – 15. The stratum level indicates the relative accuracy of the local clock compared to the clocks of other SNTP servers in the network. One is most accurate. Fifteen is least accurate.

If **-s** is not specified or an invalid value is specified, the default stratum level will be 1.

Note: The SNTP server always responds to client requests (unicast mode) whether the **-b**, **-m**, or both start options are specified.

Examples

Sample SNTPD debug output

```
Tue Apr  2 15:26:14 2002 Writing PID to file /etc/sntpd.pid
Tue Apr  2 15:26:14 2002 EZZ9602I SNTP server initializing
Tue Apr  2 15:26:14 2002 Initializing signal handling
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGINT
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGTERM
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGABND
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGABRT
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGQUIT
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGHUP
Tue Apr  2 15:26:14 2002 Set sigaction of signal SIGTTOU
Tue Apr  2 15:26:14 2002 Initializing MVS command handling
Tue Apr  2 15:26:14 2002 Initializing pthread for MVS command
Tue Apr  2 15:26:14 2002 Initializing UDP socket(s)
Tue Apr  2 15:26:15 2002 SNTP port was set to 123
Tue Apr  2 15:26:15 2002 Bound to address: 9.67.2.1
Tue Apr  2 15:26:15 2002 Bound to address: 9.67.115.15
Tue Apr  2 15:26:15 2002 Bound to address: 9.67.2.2
Tue Apr  2 15:26:15 2002 Bound to address: 0.0.0.0
Tue Apr  2 15:26:15 2002 Initializing pthread for multicast/broadcast
Tue Apr  2 15:26:15 2002 Initializing pthread for unicast
Tue Apr  2 15:26:15 2002 EZZ9600I SNTP server ready
Tue Apr  2 15:28:15 2002 Sending NTP message to multicast address 224.0.1.1
Tue Apr  2 15:30:15 2002 Sending NTP message to multicast address 224.0.1.1
```

Chapter 9. Browsing and searching syslog daemon files and archives

You can use the syslogd browse and search tool that is generally referred to as the syslogd browser to browse and search syslog daemon files and archives. See the tutorial and help panels of the syslogd browser for full and detailed information on its use.

Prerequisite: See syslogd information in z/OS Communications Server: IP Configuration Reference for details about preparing your TSO/ISPF environment for using the syslogd browser.

Syslogd browser

The syslogd browser is a TSO/ISPF application that you can use to search and browse the following syslogd message locations:

- The active UNIX files that syslogd currently is writing to.
- Syslogd MVS archive data sets that have been created with the syslogd archival function. See syslogd information in z/OS Communications Server: IP Configuration Reference for details about enabling the syslogd archival function.

Input to the syslogd browser is the name of a syslogd configuration file or data set.

Rules:

- You must use the syslogd browser on a z/OS system that has access to the active syslogd UNIX files as well as the syslogd archive data sets.
- If UNIX file systems and MVS data sets are accessible from all logical partitions (LPARs) in a z/OS sysplex, you can use the syslogd browser on any of those LPARs to access the syslogd data on any other LPAR in the z/OS sysplex; otherwise, you must use the syslogd browser on the individual LPARs to view syslogd data from each of those LPARs.
- The TSO user ID that is using the syslogd browser must have the permissions that are required to read the configuration file, access the log files, and access the archive data sets.

Starting the syslogd browser

You are able to browse syslog daemon files and archives with the syslogd browser.

To start the syslogd browser, do the following steps:

1. Start the syslogd browser from ISPF in the way that you set up. See syslogd in z/OS Communications Server: IP Configuration Reference for more information.
2. **Optional:** Enter values for the following syslogd browser options.

   Tip: You can change the values if you do not want to use the default values.

   - **Recall migrated data sets**
   - **Maximum hits to display**
   - **Maximum file archives**
   - **Display start date/time**

- **Display active files only**

  For more information about syslogd browser options, see “Syslogd browser start panel option descriptions.”

3. Enter the file or data set name of syslogd configuration, or select one from the list of the configuration files or data sets that you used.

   The default location for the syslogd configuration is `/etc/syslogd.conf`. You can specify a different file name or MVS data set to use instead. The last 10 configuration files or data set names are saved. You can reuse them by selecting them with an S line command from the list of the configuration files or data sets that you used.

   Result: When you enter or select a file or data set name, the syslogd browser reads the specified syslogd configuration and collects information about active UNIX files and available archives. For large syslogd configurations, this process might take a few seconds.

4. **Optional:** You can use line commands on each recently used syslogd configuration file or data set. For more information about line commands, see “Syslogd browser start panel line commands” on page 999.
5. Press ENTER to continue or the END PF key to exit without a selection.

Example:

When you start the browser, you can change some options and select a syslogd configuration file or data set as shown in the following example.

```
*------------------------- z/OS CS Syslogd Browser ----------- Row 1 to 7 of 7
Command ===>                                                   Scroll ===> PAGE

Enter syslogd browser options
  Recall migrated data sets ==> YES    (Yes/No) Recall data sets or not
  Maximum hits to display   ==> 200    (1-99999) Search results to display
  Maximum file archives     ==> 30     (0-400) Days to look for file archives
  Display start date/time   ==> YES    (Yes/No) Retrieve start date/time
  Display active files only ==> NO     (Yes/No) Active files only, no archives

Enter file or data set name of syslogd configuration, or select one from below:

  File/DS Name ==> 'user1.tcpcs.tcpparms(syslogt)'

Press ENTER to continue or the END PF key to exit without a selection

Line commands: S Select, R Remove from list, B Browse content, E Edit content

Cmd Recently used syslogd configuration file or data set name
--- ---------------------------------------------------------------------------
    'user1.tcpcs.tcpparms(syslogt)'
    'user1.tcpcs.tcpparms(syslogn)'
    'user1.tcpcs.tcpparms(sysltom)'
    tcpcs.tcpparms(test)
    tcpcs.tcpparms(syslogt)
    /etc/syslog.test
    /etc/syslog.alfred.conf
******************************** Bottom of data ********************************
```

Syslogd browser start panel option descriptions

You can use the following syslogd browser options to browse syslog daemon files and archives.

`Recall migrated data sets`
Configures the syslogd browser to allocate MVS data sets that are currently migrated to Level 1 or Level 2. The value YES enables the syslogd browser to

initiate a recall when such data sets are being accessed. The value NO makes such migrated data sets unavailable to the browser. The default value is NO.

Maximum hits to display
Sets an upper limit for how many hits you want displayed as the result of a search operation. You can change this value on the search panel itself. Valid values are in the range 1 - 99999. The default value is 200.

Maximum file archives
The syslogd browser supports accessing UNIX file names that include percent signs (%). %d indicates a 2-digit day, %m indicates a 2-digit month, %y indicates a 2-digit year, and %Y indicates a 4-digit year. When such file names are in use, you are likely to have some form of automation that sends a SIGHUP signal to syslogd immediately after midnight. The signal causes syslogd to close the file that was created yesterday and create a new file with today's date. The syslogd browser supports locating such files as long as they stay in the directory that they originally were created in. If your automation moves them to another location, the syslogd browser does not have enough information available to determine their location.

Use this option to set an upper limit for how many previous days archive files are to be searched. Valid values are in the range 0 - 99. If you do not use percent signs (%) in your syslogd file names, then set this option to 0. The default value is 30 days.

Display start date/time
Determines whether the syslogd browser retrieves the start date and time of each UNIX file and MVS data set. If the start date and time is retrieved, it is shown in the overview panels for each of the files and data sets that are available. For best usability, specify YES for this option. If your syslogd configuration is large (many UNIX file destinations) and you keep many data-set archive generations available, you can set this option to NO to eliminate the overhead of opening and reading the first message in each of these files and data sets to retrieve the starting date and time.

Display active files only
Set this option to NO if you intend to browse or search only in the currently active z/OS UNIX files. If you select NO, archive file or data set information is not retrieved. If your syslogd configuration is large (has many UNIX file destinations) and you keep many data-set archive generations available, you can set this option to NO to save the overhead of collecting information about the archives.

Syslogd browser start panel line commands

You can use the following line commands on each recently used syslogd configuration file or data set.

S - Select
Selects the syslogd configuration file or data set for processing.

R - Remove from list
Removes the syslogd configuration file or data set from the list.

B - Browse content
Displays the syslogd configuration file or data set in a browser window.

E - Edit content
Lets you edit the content of the syslogd configuration file or data set.

Browsing syslogd files and archives

After you start the syslogd browser, you can browse syslogd files and archives in the panel that displays the list of active UNIX files that syslogd is writing to.

Perform the following steps to list all syslogd rules that have been configured with a UNIX file name destination:

1. Press Enter in the panel that you start in “Starting the syslogd browser” on page 997.
2. **Optional:** Select one of the following primary options:

   Tip: You can choose to work with the list of files without using these options.

   - **1 Change current syslogd configuration file and/or options**
   - **2 Guide me to a possible syslogd destination**
   - **3 Clear guide-me hits (indicated by ==> in the Cmd column)**
   - **4 Search across all active syslogd files**

   For detailed information about the primary options, see “Syslogd browser display panel option descriptions” on page 1001.
3. **Optional:** You can use line commands on each destination entry that was found in the syslogd configuration. For more information about line commands, see “Syslogd browser display panel line commands” on page 1001.
4. Press ENTER to select an entry or the END PF key to exit the syslogd browser

Example:

The following example lists all syslogd rules that have been configured with a UNIX file name destination. Any syslogd rules that use /dev destinations (/dev/console, /dev/operlog, and so on), user IDs, remote syslogd servers, or AF_UNIX named pipes are not included in this list.

```
*------------------------- z/OS CS Syslogd Browser ---------- Row 1 to 8 of 12
OPTION ===>                                                   Scroll ===> PAGE

  1 Change current syslogd configuration file and/or options
  2 Guide me to a possible syslogd destination
  3 Clear guide-me hits (indicated by ==> in the Cmd column)
  4 Search across all active syslogd files

Current config file ==> 'user1.tcpcs.tcpparms(syslogt)'

Press ENTER to select an entry or the END PF key to exit the syslogd browser

Line commands: B Browse, A List archives, S Search active file and archives,
               SF Search active file, SA Search archives, I File/DSN info
                                                                    Archive
Cmd Rule/Active UNIX file name                    Start Time        Type Avail.
--- --------------------------------------------- ----------------- ---- ------
    *.*                                           22 Sep 2008 00:01 GDG  3
    /var/syslog/logs/syslog.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.TCPCS*.*.*                                  Empty       N/A   SEQ  11
    /var/syslog/logs/tcpcs.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.INETD*.*.*                                  Empty       N/A   None 0
    /var/syslog/logs/inetd.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.OSNMP*.*.*                                  Empty       N/A   CLR  0
    /var/syslog/logs/osnmpd.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.PAGENT*.*.*                                 22 Sep 2008 00:01 SEQ  12
```

```
/var/syslog/logs/pagent.log
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*.FTP*.*.*                                    Empty      N/A   FILE 10
/var/syslog/logs/ftp.21.09.08.log
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*.FTP*.*.*                                    Empty      N/A   FILE 2
/var/syslog/logs/ftp.21.09.2008.log
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
*.NAMED*.*.*                                  Empty      N/A   None 0
/var/syslog/logs/named.log
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Syslogd browser display panel option descriptions

The following primary options are available for you to use.

1 Use this option to change the current syslogd configuration. You will be returned to the initial options and syslogd configuration data panel. You can use this option when you need to reset the information that the browser collects during initialization; for example, when someone is using the syslogd browser while an archive operation occurs. In that scenario, a reinitialization is necessary so that the syslogd browser can pick up information about the new archives.

2 This option guides you to the mostly likely destination for a specific set of log messages. Use this option when you do not know where a specific set of log messages is stored.

3 Use this option to clear the indicators on individual lines that remain from a previous use of option 2. The indicator is the character string ==>.

4 Use this option to perform a search operation across all active syslogd files that are listed on the panel. Archive data sets are not included in the search when you use this option.

Syslogd browser display panel line commands

You can use one of the following line commands for each destination entry that was found in the syslogd configuration:

B - Browse
Browses the specified UNIX file.

A - List archives
Lists the available archives.

S - Search active file and archives
Performs a search operation across the active UNIX files and all available archives.

SF - Search active file
Performs a search operation across the active UNIX file only.

SA - Search archives
Performs a search operation across all available archives, but not the active UNIX file.

I - File or DSN info
Displays detailed information about the file or data set.

Searching syslogd log messages

You can invoke the search function from different locations in the syslogd browser dialog. You can limit a search operation to a single active syslogd UNIX file, a single archive, or a combination of active syslogd UNIX files and associated available archives.

Perform the following steps to search syslogd log messages:

1. Enter your search options and search arguments as shown in the following example:

   ```
   *------------------------- z/OS CS Syslogd Browser ---------------------------*
   OPTION ===>

   Enter your search options.

     Case sensitive  ==> NO        (Yes/No) Are string arguments case sensitive?
     Maximum hits    ==> 5         (1-99999) Max number of hits to display
     Result DSN name ==> 'USER1.SYSLOGD.LIST'
     Result DSN UNIT ==> SYSALLDA   Unit name for allocating new result DSN
     Result DSN disp ==> 1         1:Keep, 2:Delete, 3:Display print menu

   Enter your search arguments.  All arguments will be logically ANDed.

     From date  . . .==> 2008/10/02 (yyyy/mm/dd) Search from date
     - and time . . .==> 10:50:00   (hh:mm:ss) - and time (24-hour clock)
     To date  . . . .==> 2008/10/03 (yyyy/mm/dd) Search to date
     - and time . . .==> 02:00:00   (hh:mm:ss) - and time (24-hour clock)
     User ID  . . . .==>           z/OS user ID of logging process
     Job name . . . .==>           z/OS jobname of logging process
     Rem. host name .==>
     Rem. IP address ==>
     Message tag  . .==> Pagent          Enter ? for list
     Process ID . . .==>           z/OS UNIX process ID
     String 1 . . . .==> PAPI
     String 2 . . . .==>
     String 3 . . . .==>
     String 4 . . . .==>

   Message tags are typically component names.  PID availability depends on
   options set by the logging application.  UserID and Jobnames are available
   for local messages if syslogd is started with the -u option.

   UserID, jobname, message tag, and remote host name will always be
   case insensitive.

   Press ENTER to start search or the END PF key to return with no search
   ```

   Tip: The `case sensitive` option applies to search strings 1 - 4 only. The User ID, Job name, Message tag, and Rem. host name fields are not case sensitive.

2. **Optional:** For the message tag, you can enter a message tag value to search for, or enter a question mark (?) and press the ENTER key. In that case, a selection list is displayed in which you select the message tag that is to be part of the search arguments.

   Rule: A message must match all the specified search arguments to be considered a hit.

Result: If there are many messages to search, the search might take a few seconds. A popup panel like the following example is displayed while the search is being performed:

```
+----------------------------------------+
! *------ z/OS CS Syslogd Browser ------* !
!                                        !
```

```
!        *** S E A R C H I N G ***        !
!                                         !
!   1 of 4 files/dsn processed so far     !
!      90000 lines processed so far       !
!                                         !
!       10% |**..................|        !
!                                         !
!            Please be patient.           !
!                                         !
!   Halt by pressing ATTN and enter HI    !
!                                         !
+-----------------------------------------+
```

Sample of search results:

When the search has completed, the search results are presented in a standard ISPF view panel.

```
VIEW       USER1.SYSLOGD.LIST                                    24 hits found
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 z/OS CS Syslogd Browser Search Results - Date: 2 Sep 2008 Time: 12:30:26
000002
000003 Case sensitive  . . . NO
000004 Max. number of hits . 200
000005 Syslogd Config  . . . 'user1.tcpcs.tcpparms(syslogt)'
000006 Searched files/DSNs . 4
000007     File/DSN  . . . . /var/syslog/logs/syslog.log
000008     File/DSN  . . . . USER1.SYSLOGT.SYSLOG.G0030V00
000009     File/DSN  . . . . USER1.SYSLOGT.SYSLOG.G0031V00
000010     File/DSN  . . . . USER1.SYSLOGT.SYSLOG.G0032V00
000011
000012 Search Arguments:
000013
000014     From date . . . . 2008/08/31
000015     and time. . . . .
000016     To date . . . . . 2008/09/03
000017     and time. . . . .
000018     User ID . . . . .
000019     Job name  . . . .
000020     Remote host name.
000021     Remote IP addr. .
000022     Message tag . . . syslogd
000023     Process ID  . . .
000024     String 1  . . . . FSUM
000025     String 2  . . . .
000026     String 3  . . . .
000027     String 4  . . . .
000028
000029 Line no. File or data set: /var/syslog/logs/syslog.log
000030 ******** ***************************************************************
000031
000032 00000001 Sep  2 00:01:00 MVS098/TCPCS    SYSLOGD  syslogd: FSUM1230 Log
000033          file /var/syslog/logs/syslog.log was created
000034
000035 00000002 Sep  2 00:01:00 MVS098/TCPCS    SYSLOGD  syslogd: FSUM1230 Log
000036           file /var/syslog/logs/pagent.log was created
```

Appendix A. SNMP capability statement

This topic includes the SNMP agent and subagents capability statement for z/OS Communications Server.

The SNMP capability statement defines the MIBs supported by the SNMP Agent, **osnmpd**, and the MIBs supported by the subagents shipped as part of z/OS Communications Server.

This information is in the z/OS UNIX file system directory /usr/lpp/tcpip/samples. The file name is mvstcpip.caps.

```
-- z/OS Communications Server SNMP agent and subagents capability
-- statement
--
--
-- Program name : IBM z/OS Communications Server
--                Capabilities ASN.1 Description file
-- Requires:      IBM z/OS Communications Server
--                Version 2 Release 1
-- Description :  Defines the MIBs supported by the SNMP Agent,
--                osnmpd, and the MIBs supported by the subagents
--                shipped as part of IBM z/OS Communications Server.
--                This file is installed in the HFS as part of the
--                product install at:
--
--                      /usr/lpp/tcpip/samples/mvstcpip.caps
--
--
 IBMTCPIPMVS-CAPS DEFINITIONS ::= BEGIN
 IMPORTS
    enterprises, MODULE-IDENTITY, Integer32, OBJECT-TYPE, Unsigned32
          FROM SNMPv2-SMI
    SnmpTagValue
          FROM SNMP-TARGET-MIB
    DisplayString, TruthValue
          FROM SNMPv2-TC
    InterfaceIndex
          FROM IF-MIB
    TOSType, Status
          FROM OSPF-MIB
    AGENT-CAPABILITIES
          FROM SNMPv2-CONF;
 ibmTcpIpMvsCaps MODULE-IDENTITY
      LAST-UPDATED "201302220000Z"
      ORGANIZATION "IBM z/OS Communications Server
                    Development"
      CONTACT-INFO
          "        Kristine Adamson
           Postal: International Business Machines Corporation
                   P.O. Box 12195
                   Dept. G51A/Bldg. 501
                   Research Triangle Park, NC 27709-2195
                   USA
              Tel: +1 919 254 7911
           E-mail: adamson@us.ibm.com"
      DESCRIPTION
          "The IBM z/OS Communications Server SNMP agent
           and subagents capabilities statement.
           Licensed Materials - Property of IBM
           5650-ZOS Copyright IBM Corp. 1997, 2013"
      REVISION "201302220000Z"
      DESCRIPTION
             "Changes for release z/OS V2R1:
               - Updated copyright
               - Updated PRODUCT-RELEASE statements for V2R1
               - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
                 statement:
                  - Replaced ibmTCPIPmvsTcpGroup10 with
                    ibmTCPIPmvsTcpGroup11
```

```
            - Replaced ibmTCPIPmvsSystemGroup11 with
              ibmTCPIPmvsSystemGroup12
            - Replaced ibmTCPIPmvsInterfacesGroup10 with
              ibmTCPIPmvsInterfacesGroup11
            - Added VARIATIONS for IF-MIB MIB objects due to
              support for SMC-R RNIC interfaces
            - Replaced ibmTCPIPmvsPortGroup4 with
              ibmTCPIPmvsPortGroup5
       "
REVISION "201102140000Z"
DESCRIPTION
       "Changes for release z/OS V1R13:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R13
         - Corrected lines longer than 72 characters
         - Updated the VARIATION statement for the
           ifPhysAddress MIB object
       "
REVISION "201002080000Z"
DESCRIPTION
       "Changes for release z/OS V1R12:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R12
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Removed the VARIATION for the
              ipDefaultRouterPreference MIB object now that the
              TCP/IP stack supports RFC 4191
            - Added a VARIATION for the inetCidrRouteMetric1 MIB
              object for indirect routes created by router
              advertisements.
            - Updated the VARIATIONs for the
              ifCounterDiscontinuityTime and the
              ipSystemStatsDiscontinuityTime MIB objects
              since they are now set to a non-zero value when the
              interface is defined.
            - Added a VARIATION for the ipAdEntNetMask MIB
              object for DVIPA IP addresses on target stacks.
       "
REVISION "200809240000Z"
DESCRIPTION
       "Changes for release z/OS V1R11:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R11
         - A value of random(6) is now supported for
           the ipAddressOrigin object."
REVISION "200803010000Z"
DESCRIPTION
       "Changes for release z/OS V1R10:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R10
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsInterfacesGroup9 with
              ibmTCPIPmvsInterfacesGroup10
            - Added the ibmTCPIPmvsIfNotificationGroup
            - Changed the IF-MIB and Ether-like MIB capabilities
            - Support for the following MIB modules was upgraded
              from an IETF internet draft version to the
              RFC version:
                - IP-MIB
                - IP-FORWARD-MIB
                - TCP-MIB
                - UDP-MIB"
REVISION "200606280000Z"
DESCRIPTION
       "Changes for release z/OS V1R9:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R9
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsDVIPAGroup5 with
              ibmTCPIPmvsDVIPAGroup6
            - Replaced ibmTCPIPmvsRoutingGroup2 with
              ibmTCPIPmvsRoutingGroup3"
REVISION "200506210000Z"
DESCRIPTION
       "Changes for release z/OS V1R8:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R8
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsTcpGroup9 with
              ibmTCPIPmvsTcpGroup10
            - Replaced ibmTCPIPmvsSystemGroup10 with
              ibmTCPIPmvsSystemGroup11
            - Replaced ibmTCPIPmvsInterfacesGroup8 with
              ibmTCPIPmvsInterfacesGroup9
            - Replaced ibmTCPIPmvsDVIPAGroup4 with
              ibmTCPIPmvsDVIPAGroup5
         - Removed the ibmTcpIpMvsSlapmCaps statement for the
           Service Level Agreement subagent, pagtsnmp.
           To monitor Network Service Level Agreement
           Performance data, use the SNMP subagent, nslapm2,
           which supports the data defined in the
           NETWORK-SLAPM2-MIB module."
REVISION "200501110000Z"
DESCRIPTION
       "Changes for release z/OS V1R7:
         - Updated copyright
         - Updated PRODUCT-RELEASE statements for V1R7
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsSystemGroup9 with
              ibmTCPIPmvsSystemGroup10
            - Replaced ibmTCPIPmvsInterfacesGroup7 with
              ibmTCPIPmvsInterfacesGroup8
            - Replaced ibmTCPIPmvsPortGroup3 with
              ibmTCPIPmvsPortGroup4
            - Replaced ibmTCPIPmvsDVIPAGroup3 with
              ibmTCPIPmvsDVIPAGroup4
            - Added ibmTCPIPmvsUdpGroup4
            - Replaced ibmTCPIPmvsTcpGroup8 with
              ibmTCPIPmvsTcpGroup9"
REVISION "200402100000Z"
DESCRIPTION
       "Changes in this revision
         - Updated copyright
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsDVIPAGroup2 with
              ibmTCPIPmvsDVIPAGroup3
            - Replaced ibmTCPIPmvsTcpGroup7 with
              ibmTCPIPmvsTcpGroup8.
            - Replaced ibmTCPIPmvsSystemGroup8 with
              ibmTCPIPmvsSystemGroup9
            - Replaced ibmTCPIPmvsInterfacesGroup6 with
              ibmTCPIPmvsInterfacesGroup7
            - Added ibmTCPIPmvsRoutingGroup2
            - Updated support for the IP-MIB, the IP-FORWARD-MIB
              and the TCP-MIB
            - Replaced ibmTCPIPmvsOsaExpGroup with
              ibmTCPIPmvsOsaExpGroup2 and
              ibmTCPIPmvsOsaExpGroupOld
            - Added VARIATION statements for IF-MIB objects:
               - ifInBroadcastPkts/ifHCInBroadcastPkts
               - ifOutBroadcastPkts/ifHCOutBroadcastPkts
               - ifPhysAddress"
REVISION "200302270000Z"
DESCRIPTION
       "Changes in this revision
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Updated the inetNetToMediaLastUpdated variation.
            - Replaced ibmTCPIPmvsTcpGroup6 with
              ibmTCPIPmvsTcpGroup7 and ibmTCPIPmvsTcpGroupOld."
REVISION "200301080000Z"
DESCRIPTION
       "Changes in this revision
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsPortGroup2 with
              ibmTCPIPmvsPortGroup3 and
              ibmTCPIPmvsPortGroupOld
            - The support for the IP-MIB, IP-FORWARD-MIB, and
              TCP-MIB is now based on the IP version-neutral
              IETF internet drafts.  These drafts support
              both IPv4 and IPv6 data."
```

```
REVISION "200212180000Z"
DESCRIPTION
       "Changes in this revision
         - ibmTcpIpMvsAgtCaps capabilities extended to support
           transportDomainUdpIpv4 and transportDomainUdpIpv6
           for tAddress/tDomain pairs as described in
           RFC 3419"
REVISION "200209130000Z"
DESCRIPTION
       "Changes in this revision
         - Add new subagent, nslapm2, for NETWORK-SLAPM2-MIB to
           monitor  Network Service Level Agreement
           Performance.
         - Updated PRODUCT-RELEASE for V1R5
         - Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
           statement:
            - Replaced ibmTCPIPmvsInterfacesGroup5 with
              ibmTCPIPmvsInterfacesGroup6
            - Replaced ibmTCPIPmvsDVIPAGroup with
              ibmTCPIPmvsDVIPAGroup2
            - Replaced ibmTCPIPmvsSystemGroup7 with
              ibmTCPIPmvsSystemGroup8
         - Added the ibmMvsTN3270SaCaps AGENT-CAPABILITIES
           statement for the SNMP TN3270 Subagent"
REVISION "200203110000Z"
DESCRIPTION
       "Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
        statement:
         - ipRoutingDiscards not supported
         - icmpOutRedirects variation"
REVISION "200103160000Z"
DESCRIPTION
       "Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
        statement in regards to the following MIB groups:
         - Replaced ibmTCPIPmvsTcpGroup5 with
           ibmTCPIPmvsTcpGroup6
         - Added ibmTCPIPmvsIpGroup
         - Added support for ifVHCPacketGroup from RFC 2233
         - ifAdminstatus no longer supported for enabling/
           disabling an OSA ATM physical port.
         - Replaced ibmTCPIPmvsSystemGroup6 with
           ibmTCPIPmvsSystemGroup7
         - Added ibmTCPIPmvsOsaExpGroup
         - Replaced ibmTCPIPmvsInterfacesGroup4 with
           ibmTCPIPmvsInterfacesGroup5
         - Added ibmTCPIPmvsDVIPAGroup
         - Added ibmTCPIPmvsDVIPANotificationGroup
         - Added ibmTCPIPmvsSystemNotificationGroup
        Corrected name of ibmAgentCapabilities to
        to ibmAgentCaps"
REVISION "200003010000Z"
DESCRIPTION
       "Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
        statement in regards to the following MIB groups:
         - Replaced ibmTCPIPmvsSystemGroup5 with
           ibmTCPIPmvsSystemGroup6
         - Replaced ibmTCPIPmvsAtmLeGroup with
           ibmTCPIPmvsAtmLeGroup2"
REVISION "200002090000Z"
DESCRIPTION
       "Changed product name from SecureWay Communications
        Server for OS/390 to IBM Communications Server for
        OS/390"
REVISION "200002030000Z"
DESCRIPTION
       "Modified the ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
        statement in regards to the following MIB objects:
         - ipAdEntReasmMaxSize
         - ifInNUcastPkts
         - ifOutNUcastPkts
         - ifSpecific."
REVISION "200001240000Z"
DESCRIPTION
       "Modified the ibmTcpIpMvsSlapmCaps AGENT-CAPABILITIES
        statement to reflect the implementation of a newer
        version of the SLAPM-MIB."
REVISION "9911160000Z"
DESCRIPTION
       "Changes in this revision
```

```
             - Added ibmTCPIPmvsTcpGroup5
             - Added ibmTCPIPmvsUdpGroup3
             - Added support for EtherLike-MIB in RFC2665"
     REVISION "9908310000Z"
     DESCRIPTION
            "Changes in this revision
             - Removed variations that restricted the use of UTF8
               characters for SnmpAdminString objects.
             - Added support for snmpNotifyFilterGroup
             - Added support for inform type notifications"
     REVISION "9908060000Z"
     DESCRIPTION
            "Modified the ibmTcpIpMvsSlapmCaps AGENT-CAPABILITIES
             statement to reflect the implementation of a newer
             version of the SLAPM-MIB."
     REVISION "9907010000Z"
     DESCRIPTION
            "Changes in this revision
             - Added ibmTCPIPmvsInterfacesGroup4
             - Added ibmTCPIPmvsPortGroup2
             - Added ibmTCPIPmvsAtmSupportGroup4"
     REVISION "9903300000Z"
     DESCRIPTION
            "Changes in this revision
             - Added ibmTCPIPmvsTcpGroup4
             - Added ibmTCPIPmvsAtmSupportGroup3"
     REVISION "9902150000Z"
     DESCRIPTION
            "Changes in this revision
             - Changed product name from eNetwork Communications
               Server to SecureWay Communications Server"
     REVISION "9811240000Z"
     DESCRIPTION
            "Changes in this revision
             - Added statement to document the MIB support
               provided by the new Service Level Agreement
               subagent, pagtsnmp.
             - Added ibmTCPIPmvsSystemGroup5"
     REVISION "9807130000Z"
     DESCRIPTION
            "Changes in this revision
             - Added SNMPV3 support
             - Removed support for SNMPv2-USEC-MIB"
     REVISION "9806120000Z"
     DESCRIPTION
            "Added OSPF-MIB support"
     REVISION "9805120000Z"
     DESCRIPTION
            "Changes in this revision
             - Added ibmTCPIPmvsSystemGroup4
             - Added ibmTCPIPmvsInterfacesGroup3"
     REVISION "9804150000Z"
     DESCRIPTION
            "Added IPOA-MIB support"
     REVISION "9803050000Z"
     DESCRIPTION
            "Changes in this revision
             - Added copyright
             - Changed CONTACT-INFO"
::= { ibmAgentCaps 7 }
ibm                  OBJECT IDENTIFIER ::= { enterprises 2 }
ibmAgentCaps         OBJECT IDENTIFIER ::= { ibm 11 }
ibmTcpIpMvsAgtCaps AGENT-CAPABILITIES
   PRODUCT-RELEASE  "IBM z/OS Communications Server
                     Version 2 Release 1 SNMP Agent"
   STATUS           current
   DESCRIPTION      "IBM z/OS Communications Server Agent"
   SUPPORTS             SNMPv2-MIB       -- RFC 1907
       INCLUDES         { systemGroup, snmpGroup, snmpSetGroup,
                          snmpBasicNotificationsGroup,
                          snmpCommunityGroup }
       VARIATION        coldStart
           DESCRIPTION  "A coldStart trap is generated on all
                        reboots."
   SUPPORTS             DPI20-MIB        -- RFC 1592
       INCLUDES         { dpiGroup }
       VARIATION        dpiPathNameForUnixStream
           DESCRIPTION  "This object was added to the dpiMib
                         defined by RFC1592 in order to support
                         AF_UNIX DPI connections. Its SMI
                         definition is:
                         dpiPathNameForUnixStream OBJECT-TYPE
                         SYNTAX      DisplayString
                         MAX-ACCESS  read-only
                         STATUS      current
                         DESCRIPTION
                          'The full path name for a connection via an
                           AF_UNIX stream connection. The empty value
                           means the agent has no DPI AF_UNIX support.'
                         ::= { dpiPort 3 }
                         Replace the single quotes with double
                         quotes in the DESCRIPTION of this object
                         when compiling."
   -- This MIB was posted to the agentx mailing list in the IETF.
   -- A copy of this MIB is installed as samib.mi2 in HFS at
   -- /usr/lpp/tcpip/samples as part of installing the
   -- IBM z/OS Communications Server.
   SUPPORTS             SUBAGENT-MIB
       INCLUDES          { saTableGroup, saTreeGroup }
   SUPPORTS             SNMP-FRAMEWORK-MIB
       INCLUDES          { snmpEngineGroup }
   SUPPORTS             SNMP-MPD-MIB
       INCLUDES          { snmpMPDGroup }
   SUPPORTS             SNMP-TARGET-MIB
       INCLUDES          { snmpTargetBasicGroup,
                           snmpTargetResponseGroup,
                           snmpTargetCommandResponderGroup }
       VARIATION        snmpTargetAddrTagList
           SYNTAX       SnmpTagValue
           DESCRIPTION  "Only single-value tagList is supported"
   SUPPORTS             SNMP-NOTIFICATION-MIB
       INCLUDES          { snmpNotifyGroup,
                           snmpNotifyFilterGroup }
   SUPPORTS             SNMP-USER-BASED-SM-MIB
       INCLUDES          { usmMIBBasicGroup }
   SUPPORTS             SNMP-VIEW-BASED-ACM-MIB
       INCLUDES          { vacmBasicGroup }
       VARIATION        vacmContextName
           SYNTAX       DisplayString (SIZE(0..32))
           DESCRIPTION  "Only the null context is supported"
   SUPPORTS             SNMP-COMMUNITY-MIB  -- RFC 3584
       INCLUDES          { snmpCommunityTableGroup }
   ::= { ibmTcpIpMvsCaps 1 }
ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
   PRODUCT-RELEASE  "IBM z/OS Communications Server
                     Version 2 Release 1 TCP/IP Subagent"
   STATUS           current
   DESCRIPTION      "IBM z/OS Communications Server
                     TCP/IP DPI Subagent"
   -- Our enterprise specific MIB. Its SMI definition, mvstcpip.mi2,
   -- is shipped with IBM z/OS Communications Server and
   -- installed in the HFS at: /usr/lpp/tcpip/samples
   SUPPORTS       IBMTCPIPMVS-MIB
       INCLUDES {  ibmTCPIPmvsPingGroup2,
                   ibmTCPIPmvsSystemGroup12,
                   ibmTCPIPmvsTcpGroup11,
                   ibmTCPIPmvsTcpGroupOld,
                   ibmTCPIPmvsUdpGroup3,
                   ibmTCPIPmvsUdpGroup4,
                   ibmTCPIPmvsInterfacesGroup11,
                   ibmTCPIPmvsPortGroup5,
                   ibmTCPIPmvsPortGroupOld,
                   ibmTCPIPmvsRoutingGroup3,
                   ibmTCPIPmvsRoutingGroup,
                   ibmTCPIPmvsIpGroup,
                   ibmTCPIPmvsAtmSupportGroup4,
                   ibmTCPIPmvsAtmNotificationGroup,
                   ibmTCPIPmvsAtmLeGroup2,
                   ibmTCPIPmvsOsaExpGroup2,
                   ibmTCPIPmvsOsaExpGroupOld,
                   ibmTCPIPmvsDVIPAGroup6,
                   ibmTCPIPmvsDVIPANotificationGroup,
                   ibmTCPIPmvsSystemNotificationGroup }
       VARIATION        osasfChannelTable
           DESCRIPTION  "The OSA-Express ATM155 adapter management
                         data has been moved to the
                         osaexpChannelTable as of V1R2. Therefore,
                         the OSA-Express ATM155 values for the
```

```
                  following MIB objects will never be set:
                    - ibmMvsAtmOsasfChannelType
                    - ibmMvsAtmOsasfChannelSubType
                    - ibmMvsAtmOsasfChannelHwModel"
VARIATION        osaexpEthPortTable
     DESCRIPTION  "The table entries are indexed by the
                   interface index of either, an interface
                   defined by the DEVICE profile statement,
                   or a dynamically generated OSA-Express
                   QDIO port interface."
VARIATION        osaexpEthSnaTable
     DESCRIPTION  "The table entries are indexed by the
                   interface index of either, an interface
                   defined by the DEVICE profile statement,
                   or a dynamically generated OSA-Express
                   QDIO port interface."
VARIATION        ibmMvsDVIPARangeConfMoveable
    ACCESS       read-only
    DESCRIPTION  "This object is supported for read-only
                  access."
VARIATION        ibmMvsDVIPARangeConfStatus
    SYNTAX       INTEGER { active(1) }
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPARangeConfTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access and the only
                  value supported is active(1)."
VARIATION        ibmMvsDVIPADistConfStatus
    SYNTAX       INTEGER { active(1) }
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPADistConfTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access and the only
                  value supported is active(1)."
VARIATION        ibmMvsDVIPADistConfTimedAffinity
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPADistConfTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access."
VARIATION        ibmMvsDVIPADistConfSplxPortsEn
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPADistConfTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access."
VARIATION        ibmMvsDVIPADistConfDistMethod
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPADistConfTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access."
VARIATION        ibmMvsDVIPADistConfIntfName
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPADistConfTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access."
VARIATION        ibmMvsDVIPARangeConfigMoveable
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPARangeConfigTable via an snmp set
                  command to this object.  The object is
                  supported for read-only access."
VARIATION        ibmMvsDVIPARangeConfigIntfName
    ACCESS       read-only
    DESCRIPTION  "This implementation does not support dynamic
                  row creation of a conceptual row in the
                  ibmMvsDVIPARangeConfigTable via an snmp set
                  command to this object.  The object is
```

```
                          supported for read-only access."
        VARIATION        ibmMvsDVIPARangeConfigStatus
            SYNTAX       INTEGER { active(1) }
            ACCESS       read-only
            DESCRIPTION  "This implementation does not support dynamic
                          row creation of a conceptual row in the
                          ibmMvsDVIPARangeConfigTable via an snmp set
                          command to this object.  The object is
                          supported for read-only access and the only
                          value supported is active(1)."
    SUPPORTS             IF-MIB   -- RFC 2233
        INCLUDES         { ifGeneralInformationGroup,
                           ifStackGroup2,
                           ifPacketGroup,
                           ifHCFixedLengthGroup,
                           ifVHCPacketGroup}
        VARIATION        ifTable
            DESCRIPTION  "This implementation creates dynamic entries
                          in the table for OSA-Express QDIO Ethernet
                          ports associated with IPAQENET/IPAQENET6
                          interfaces defined by the INTERFACE profile
                          statement.
                          For RNIC interfaces, the following counter
                          MIB objects are not supported and will
                          always be set to a value of zero:
                            ifInDiscards
                            ifInErrors
                            ifInUnknownProtos
                            ifOutDiscards
                            ifOutErrors
                            ifOutQLen
                          RNIC interfaces can be identified by the
                          value of rnic(39) set in MIB object
                          ibmMvsIfType for an interface.
                         "
        VARIATION        ifType
            DESCRIPTION  "A value of propVirtual(53) is set for
                          interfaces defined by a DEVICE profile
                          statement, or for interfaces
                          representing an OSA-Express QDIO Ethernet
                          port."
        VARIATION        ifMtu
            DESCRIPTION  "For ATM LAN Emulation interfaces configured
                          for token ring, this value is the maximum
                          data frame size minus 54 octets for
                          encapsulation. For ATM LAN Emulation
                          interfaces not configured for token ring,
                          this value is the maximum dataframe size."
        VARIATION        ifPhysAddress
            DESCRIPTION  "Only supported for the following interface
                          types when the interface is active:
                            - ATM
                            - HCH
                            - LCS Ethernet, Token Ring, FDDI
                            - OSA-Express Ethernet, Token Ring
                            - RNIC
                          For OSA-Express Ethernet interfaces, the
                          value will be one of the following:
                            - a physical MAC address
                            - a Virtual MAC address specified by the
                              customer
                            - a Virtual MAC address generated by the
                              OSA-Express feature
                          For HiperSockets IQDX interfaces, the value
                          will be a HiperSockets-generated Virtual
                          MAC address
                          For RNIC interfaces, the value will be
                          the Virtual MAC address generated by
                          the VTAM DLC layer.
                         "
        VARIATION        ifAdminStatus
            SYNTAX       INTEGER { up(1), down(2) }
            DESCRIPTION  "Test mode (testing(3)) not supported. The
                          set operation is not supported for the
                          following interfaces:
                           - loopback
                           - Virtual IP Address (VIPA)
                           - dynamically created OSA-Express port
                          This object reflects the desired state of
```

```
                  an interface. If a START command has been
                  invoked for an interface,
                  ifAdminStatus will be set to up(1). If an
                  interface has never been started, or if
                  a STOP command has been invoked for an
                  interface, ifAdminStatus will be set to
                  down(2)."
VARIATION        ifOperStatus
    SYNTAX       INTEGER { up(1), down(2) }
    DESCRIPTION  "Information limited to up or down. Do not
                  support testing(3), unknown(4), dormant(5),
                  notPresent(6), nor lowerLayerDown(7).
                  For dynamically created OSA-Express port
                  table entries, the value of this object
                  will be set to up(1) if any interface
                  associated with the port is up (active).
                  The value of this object will be set to
                  down(2) if all interfaces associated with
                  the port are down (inactive)."
VARIATION        ifLastChange
    DESCRIPTION  "Use time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime represents time relative to the
                  agents IPL not TCP/IPs."
VARIATION        ifInUcastPkts
    DESCRIPTION  "For RNIC interfaces, the value in this
                  MIB object is the number of Remote Direct
                  Memory Access (RDMA) work elements processed
                  for inbound data over the interface.
                 "
VARIATION        ifInNUcastPkts
    DESCRIPTION  "This implementation does not maintain this
                  object.  The value of the object will
                  always be zero."
VARIATION        ifOutUcastPkts
    DESCRIPTION  "For RNIC interfaces, the value in this
                  MIB object is the number of RDMA over
                  Converged Ethernet (ROCE) post
                  operations for transferring data across
                  this interface.
                 "
VARIATION        ifOutNUcastPkts
    DESCRIPTION  "This implementation does not maintain this
                  object.  The value of the object will
                  always be zero."
VARIATION        ifSpecific
    DESCRIPTION  "This implementation does not maintain this
                  object.  The value of the object will
                  always be 0.0."
VARIATION        ifXTable
    DESCRIPTION  "This implementation creates dynamic entries
                  in the table for OSA-Express QDIO Ethernet
                  ports associated with IPAQENET/IPAQENET6
                  interfaces defined by the INTERFACE profile
                  statement.
                  For RNIC interfaces, the following counter
                  MIB objects are not supported and will
                  always be set to a value of zero:
                    ifHCInMulticastPkts
                    ifHCInBroadcastPkts
                    ifHCOutMulticastPkts
                    ifHCOutBroadcastPkts
                  RNIC interfaces can be identified by the
                  value of rnic(39) set in MIB object
                  ibmMvsIfType for an interface.
                 "
VARIATION        ifHCInUcastPkts
    DESCRIPTION  "For RNIC interfaces, the value in this
                  MIB object is the number of Remote Direct
                  Memory Access (RDMA) work elements processed
                  for inbound data over the interface.
                 "
VARIATION        ifInBroadcastPkts
    DESCRIPTION  "Only supported for the following interface
                  types:
                   - LCS Ethernet, Token Ring, FDDI
                   - MPCIPA Ethernet, Token Ring,
                            HiperSockets"
VARIATION        ifHCOutUcastPkts
```

```
            DESCRIPTION  "For RNIC interfaces, the value in this
                          MIB object is the number of RDMA over
                          Converged Ethernet (ROCE) post
                          operations for transferring data across
                          this interface.
                         "
        VARIATION        ifOutBroadcastPkts
            DESCRIPTION  "Only supported for the following interface
                          types:
                           - LCS Ethernet, Token Ring, FDDI
                           - MPCIPA Ethernet, Token Ring,
                                    HiperSockets"
        VARIATION        ifHCInBroadcastPkts
            DESCRIPTION  "Only supported for the following interface
                          types:
                           - LCS Ethernet, Token Ring, FDDI
                           - MPCIPA Ethernet, Token Ring,
                                    HiperSockets"
        VARIATION        ifHCOutBroadcastPkts
            DESCRIPTION  "Only supported for the following interface
                          types:
                           - LCS Ethernet, Token Ring, FDDI
                           - MPCIPA Ethernet, Token Ring,
                                    HiperSockets"
        VARIATION        ifLinkUpDownTrapEnable
            SYNTAX       INTEGER { enabled(1), disabled(2) }
            DESCRIPTION  "A value of enabled(1), is not supported for
                          interface table entries which represent a
                          dynamically generated OSA-Express QDIO
                          Ethernet port."
        VARIATION        ifPromiscuousMode
            ACCESS        read-only
            DESCRIPTION  "Write access is not required, nor supported."
        VARIATION        ifCounterDiscontinuityTime
            DESCRIPTION  "Use time that TCP/IP was started instead of
                          sysUpTime to calculate this value, since
                          sysUpTime represents time relative to the
                          agents IPL not TCP/IPs.  This value is
                          set for the following events:
                           - when an interface is first defined.
                           - when an existing interface is deleted from
                             and then defined again to the stack
                           - when certain errors occur on an interface.
                         "
        VARIATION        ifStackTable
            DESCRIPTION  "This implementation creates dynamic entries
                          in the table for OSA-Express QDIO Ethernet
                          ports associated with IPAQENET/IPAQENET6
                          interfaces defined by the INTERFACE profile
                          statement."
        VARIATION        ifStackStatus
            SYNTAX       INTEGER { active(1) }
            ACCESS       read-only
            DESCRIPTION  "Write access is not required, nor supported.
                          Only one enumerated values for the RowStatus
                          textual convention is supported."
        VARIATION        ifStackLastChange
            DESCRIPTION  "Not supported"
    SUPPORTS             IP-MIB   -- RFC 4293
        INCLUDES         { ipGroup, icmpGroup,
                           ipSystemStatsGroup,   ipAddressGroup,
                           ipNetToPhysicalGroup, ipDefaultRouterGroup,
                           icmpStatsGroup,
                           ipSystemStatsHCOctetGroup,
                           ipSystemStatsHCPacketGroup,
                           ipv6GeneralGroup2,
                           ipv6IfGroup,
                           ipAddressPrefixGroup,
                           ipLastChangeGroup }
        VARIATION        ipReasmTimeout
            ACCESS        read-write
            DESCRIPTION  "This implementation of the TCP/IP
                          protocols allows this configuration
                          parameter to be changed."
        VARIATION        ipNetToMediaIfIndex
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaPhysAddress
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaNetAddress
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipNetToMediaType
            ACCESS        read-only
            DESCRIPTION  "Write access not supported."
        VARIATION        ipAddrTable
            DESCRIPTION  "Not all existing instances can be supported
                          because the index is an IP address and
                          the TCP/IP stack allows the same IP
                          address to be defined for multiple
                          interfaces."
        VARIATION        ipAdEntNetMask
            DESCRIPTION  "For dynamic VIPA (DVIPA) IP addresses on
                          target stacks of the Sysplex Distributor
                          function, a value of 0 will be set for this
                          object to indicate that the IP address is
                          not owned or advertised by the target stack.
                         "
        VARIATION        ipAdEntReasmMaxSize
            DESCRIPTION  "Since this implementation does not support
                          unique reassembly size values per interface,
                          the value for this object for all interfaces
                          will be the constant 65535."
        VARIATION        ipRoutingDiscards
            DESCRIPTION  "This implementation does not maintain this
                          object.  The value of the object will
                          always be 0."
        VARIATION        icmpOutRedirects
            DESCRIPTION  "This implementation does not send ICMP
                          Redirect messages but, since it includes
                          in this object any Redirect messages sent
                          by an application, this object may not
                          be 0."
        VARIATION        ipv6IpForwarding
            DESCRIPTION  "If an snmp set request is processed for this
                          object, the value from the set request is
                          not written to non-volatile storage.  So
                          the new value is only in effect until the
                          next set request for the object, until a
                          VARY TCPIP,,OBEYFILE commend is processed
                          that changes the value, or until the TCP/IP
                          stack is recycled."
        VARIATION        ipv6IpDefaultHopLimit
            DESCRIPTION  "Value of 0 not supported.  Supports values
                          of 1-255.
                          If an snmp set request is processed for this
                          object, the value from the set request is
                          not written to non-volatile storage.  So
                          the new value is only in effect until the
                          next set request for the object, until a
                          VARY TCPIP,,OBEYFILE commend is processed
                          that changes the value, or until the TCP/IP
                          stack is recycled."
        VARIATION        ipv6InterfaceTableLastChange
            DESCRIPTION  "Uses time that TCP/IP was started instead of
                          sysUpTime to calculate this value, since
                          sysUpTime represents time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        ipv6InterfaceReasmMaxSize
            DESCRIPTION  "The value of this MIB object will always
                          be 65535."
        VARIATION        ipv6InterfaceEnableStatus
            ACCESS        read-only
            DESCRIPTION  "The value of this MIB object will always
                          be up(1).  Write access is not supported."
        VARIATION        ipv6InterfaceForwarding
            ACCESS        read-only
            DESCRIPTION  "The value of this MIB object will always
                          be forwarding(1) since this implementation
                          does not provide per-interface control of
                          the forwarding function.  Write access
                          is not supported."
        VARIATION        ipSystemStatsDiscontinuityTime
            DESCRIPTION  "Uses time that TCP/IP was started instead of
                          sysUpTime to calculate this value, since
                          sysUpTime represents time relative to the
                          agents IPL not TCP/IPs.  This value is
                          set for the following events:
                           - when an interface is first defined.
                           - when an existing interface is deleted
                             from and then defined again to the stack
                           - when certain errors occur on an interface.
                         "
        VARIATION        ipSystemStatsRefreshRate
            DESCRIPTION  "This object will be set to the TCP/IP
                          Subagent's current cache time since a
                          management application will not see a
                          change in the counter values until the
                          cache time expires."
        VARIATION        ipAddressPrefixTable
            DESCRIPTION  "This implementation does not support
                          IPv6 entries in this table for prefixes
                          from Router Advertisements where the
                          on-link flag was 'off' and either the
                          autonomous flag was 'off' or
                          autoconfiguration of IP addresses was not
                          being performed for the interface."
        VARIATION        ipAddressPrefixOnLinkFlag
            DESCRIPTION  "This implementation does not support
                          entries in this table for which this
                          object would have a value of false(2).
                          The value of this object will be true(1)
                          for all entries."
        VARIATION        ipAddressSpinLock
            ACCESS        read-only
            DESCRIPTION  "The value of this MIB object will always
                          be 0.  Write access is not supported."
        VARIATION        ipAddressOrigin
            SYNTAX       INTEGER {
                                   other(1),
                                   manual(2),
                                   linklayer(5),
                                   random(6)
                                 }
            DESCRIPTION  "This implementation does not support the
                          value dhcp(4)."
        VARIATION        ipAddressCreated
            DESCRIPTION  "Uses time that TCP/IP was started instead of
                          sysUpTime to calculate this value, since
                          sysUpTime represents time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        ipAddressLastChanged
            DESCRIPTION  "Uses time that TCP/IP was started instead of
                          sysUpTime to calculate this value, since
                          sysUpTime represents time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        ipAddressRowStatus
            ACCESS       read-only
            DESCRIPTION  "This implementation does not support dynamic
                          row creation of a conceptual row in the
                          ipAddressTable via an snmp set
                          command to this object.  The object is
                          supported for read-only access and the only
                          value supported is active(1)."
        VARIATION        ipNetToPhysicalLastUpdated
            DESCRIPTION  "Uses time that TCP/IP was started instead of
                          sysUpTime to calculate this value, since
                          sysUpTime represents time relative to the
                          agents IPL not TCP/IPs.
                          There are some OSA adapters which maintain
                          the IPv4 ARP cache data on the adapter.
                          For entries in this table where the IPv4 ARP
                          cache data is being maintained by an OSA
                          adapter, the value for this object indicates
                          the last time the IPv4 ARP cache information
                          was retrieved by the stack from the adapter.
                          It does not necessarily mean that the IPv4
                          ARP cache data has changed."
        VARIATION        ipNetToPhysicalType
            SYNTAX       INTEGER { other(1), dynamic(3), static(4),
                                   local(5) }
            DESCRIPTION  "This implementation does not support
                          a value of invalid(2)."
        VARIATION        ipNetToPhysicalState
            SYNTAX       INTEGER {
                                   reachable(1),
                                   stale(2),
                                   delay(3),
                                   probe(4),
                                   unknown(6),
                                   incomplete(7)
                                 }
            DESCRIPTION  "This implementation does not support
                          a value of invalid(5)."
        VARIATION        ipLastChangeGroup
            DESCRIPTION  "This implementation only supports the
                          ipv6IfTableLastChange object from this
                          group."
    SUPPORTS             IP-FORWARD-MIB   -- RFC 4292
        INCLUDES         { ipForwardMultiPathGroup,
                           inetForwardCidrRouteGroup }
        VARIATION        ipForwardMask
            ACCESS        read-only
            DESCRIPTION "Write access not supported."
        VARIATION        ipForwardPolicy
            DESCRIPTION "Not used in this release. Will always return
                         a zero."
        VARIATION        ipForwardIfIndex
            ACCESS        read-only
            DESCRIPTION  "write access not supported."
        VARIATION        ipForwardType
            ACCESS        read-only
            DESCRIPTION  "write access not spported."
        VARIATION        ipForwardInfo
            ACCESS        read-only
            DESCRIPTION "write access not supported.
                         Will always return a zero"
        VARIATION        ipForwardNextHopAS
            ACCESS        read-only
            DESCRIPTION "write access not supported.
                         Will always return a zero."
        VARIATION        ipForwardMetric1
            ACCESS        read-only
            DESCRIPTION
               "An alternate routing metric  for  this  route."
        VARIATION        ipForwardMetric2
            ACCESS        read-only
            DESCRIPTION  "not supported"
        VARIATION        ipForwardMetric3
            ACCESS        read-only
            DESCRIPTION  "not supported"
        VARIATION        ipForwardMetric4
            ACCESS        read-only
            DESCRIPTION  "not supported"
        VARIATION        ipForwardMetric5
            ACCESS        read-only
            DESCRIPTION  "not supported"
        VARIATION        inetCidrRouteType
            DESCRIPTION  "This implementation does not support values
                          of other(1) and blackhole(5).  A value
                          of reject(2) will only be set for the case
                          where the interface associated with the
                          route is not active."
        VARIATION        inetCidrRouteAge
            DESCRIPTION  "This implementation does not periodically
                          verify that the route is correct, so this
                          object will only indicate the time since
                          the route was created."
        VARIATION        inetCidrRouteMetric1
            DESCRIPTION  "For IPv6 indirect routes which were created
                          because of a router advertisement, the
                          value of the object will be the preference
                          value from the router advertisement,
                          as follows:
                            1 - High
                            2 - Medium
                            3 - Low
                         "
        VARIATION        inetCidrRouteStatus
            SYNTAX       INTEGER { active(1) }
            ACCESS        read-only
            DESCRIPTION  "This implementation does not support dynamic
                          row creation of a conceptual row in the
                          inetCidrRouteTable via an snmp set
                          command to this object.  The object is
```

```
                          supported for read-only access and the only
                          value supported is active(1)."
        VARIATION        inetCidrRouteDiscards
            DESCRIPTION  "This implementation does not support
                          this object."
    SUPPORTS             TCP-MIB   -- RFC 4022
        INCLUDES         { tcpGroup,
                           tcpBaseGroup, tcpConnectionGroup,
                           tcpListenerGroup,
                           tcpHCGroup}
        VARIATION        tcpActiveOpens
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpPassiveOpens
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpAttemptFails
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpEstabResets
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpInSegs
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpOutSegs
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpRetransSegs
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpInErrs
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpOutRsts
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpHCInSegs
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpHCOutSegs
            DESCRIPTION  "Discontinuities in the value of this counter
                          can only occur at re-initialization of the
                          TCP/IP stack.  Discontinuities in the value of
                          sysUpTime represent time relative to the
                          agents IPL not TCP/IPs."
        VARIATION        tcpConnectionProcess
            DESCRIPTION  "Since this implementation does not support
                          the HOST-RESOURCES-MIB nor the
```

```
                      SYSAPPL-MIB, the value of this object will
                      always be 0."
    VARIATION        tcpListenerProcess
        DESCRIPTION  "Since this implementation does not support
                      the HOST-RESOURCES-MIB nor the
                      SYSAPPL-MIB, the value of this object will
                      always be 0."
SUPPORTS             UDP-MIB   -- RFC 4113
    INCLUDES         { udpGroup, udpBaseGroup, udpHCGroup,
                       udpEndpointGroup }
    VARIATION        udpInDatagrams
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value
                      of sysUpTime do not necessarily imply
                      discontinuities in this counter."
    VARIATION        udpNoPorts
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value
                      of sysUpTime do not necessarily imply
                      discontinuities in this counter."
    VARIATION        udpInErrors
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value
                      of sysUpTime do not necessarily imply
                      discontinuities in this counter."
    VARIATION        udpOutDatagrams
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value
                      of sysUpTime do not necessarily imply
                      discontinuities in this counter."
    VARIATION        udpHCInDatagrams
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value of
                      sysUpTime represent time relative to the
                      agents IPL not TCP/IPs."
    VARIATION        udpHCOutDatagrams
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value of
                      sysUpTime represent time relative to the
                      agents IPL not TCP/IPs."
    VARIATION        udpTable
        DESCRIPTION  "Not all existing instances can be supported
                      because the index is the local address
                      and port.  If the socket option SO_REUSEADDR
                      is specified on a setsockopt() for a UDP
                      listener, then the TCP/IP stack allows
                      more than one listener to bind to the same
                      multicast IP address and port."
    VARIATION        udpEndpointInstance
        DESCRIPTION  "This implementation sets this MIB object
                      to the value of the connection ID for the
                      UDP endpoint."
    VARIATION        udpEndpointProcess
        DESCRIPTION  "Since this implementation does not support
                      the HOST-RESOURCES-MIB nor the
                      SYSAPPL-MIB, the value of this object will
                      always be 0."
    VARIATION        udpHCInDatagrams
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value
                      of sysUpTime do not necessarily imply
                      discontinuities in this counter."
    VARIATION        udpHCOutDatagrams
        DESCRIPTION  "Discontinuities in the value of this counter
                      can only occur at re-initialization of the
                      TCP/IP stack.  Discontinuities in the value
                      of sysUpTime do not necessarily imply
                      discontinuities in this counter."
SUPPORTS             ATM-MIB      -- RFC 1695
    INCLUDES         { atmInterfaceConfGroup }
    VARIATION        atmInterfaceConfVpcs
        DESCRIPTION  "For OSA purposes this object is defined
```

```
                      as the number of active VPCs (PVCs and
                      SVCs)."
    VARIATION        atmInterfaceConfVccs
        DESCRIPTION  "For OSA purposes this object is defined
                      as the number of active VCCs (PVCs and
                      SVCs)."
    VARIATION        atmInterfaceIlmiVpi
        DESCRIPTION  "The VPI value of the VCC supporting the
                      ILMI at this ATM interface. If the values
                      of atmInterfaceVpi and atmInterfaceVci
                      are both equal to zero, than the ILMI is not
                      supported at this atm interface. Only valid
                      value is currently 0."
    VARIATION        atmInterfaceIlmiVci
        DESCRIPTION  "The VPI value of the VCC supporting the
                      ILMI at this ATM interface. If the values
                      of atmInterfaceVpi and atmInterfaceVci
                      are both equal to zero, than the ILMI is not
                      supported at this atm interface. Only valid
                      value is currently 16."
    VARIATION        atmInterfaceAddressType
        DESCRIPTION  "The type of primary ATM address configured
                      for use at this ATM interface. Only valid
                      value on current OSA is 1."
SUPPORTS             IBM3172-MIB      -- IBM 3172 MIB
    INCLUDES         { ibm3172Group }
SUPPORTS             IPOA-MIB         -- IP over ATM MIB RFC 2320
    INCLUDES         { ipoaGeneralGroup}
    VARIATION     ipoaLisTrapEnable
         DESCRIPTION  "This implementation does not support
                       this object."
    VARIATION     ipoaLisDefaultMtu
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisDefaultEncapsType
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set. Object can only
                       be llcsnap."
    VARIATION     ipoaLisInactivityTimer
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set. Smallest value
                       is 10 seconds. Default value is 300.
                       A zero continues to indicate
                       no time out in effect."
    VARIATION     ipoaLisMinHoldingTime
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisQDepth
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisMaxCalls
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisCacheEntryAge
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisRetries
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisTimeout
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set. Our default is
                       3 seconds."
    VARIATION     ipoaLisDefaultPeakCellRate
         ACCESS    read-only
         DESCRIPTION  "This implementation does not allow
                       this object to be set."
    VARIATION     ipoaLisRowStatus
         DESCRIPTION  "This implementation does not support
                       this object."
        VARIATION     ipoaLisIfMappingRowStatus
             ACCESS    read-only
             DESCRIPTION  "This implementation does not support
                           remote creation."
        VARIATION     ipoaArpClientAtmAddr
             ACCESS    read-only
             DESCRIPTION  "This implementation does not support
                           setting this object."
        VARIATION     ipoaArpClientRowStatus
             DESCRIPTION  "This implementation does not support
                           this object."
        VARIATION     ipoaArpSrvrTable
             DESCRIPTION  "This implementation does not support
                           this object."
        VARIATION     ipoaArpRemoteSrvrRowStatus
             DESCRIPTION  "This implementation does not support
                           this object."
        VARIATION     ipoaArpRemoteSrvrAdminStatus
             DESCRIPTION  "This implementation does not support
                           this object."
        VARIATION     ipoaArpRemoteSrvrOperStatus
             DESCRIPTION  "This implementation does not support
                           this object."
        VARIATION     ipoaVcNegotiatedEncapsType
             DESCRIPTION  "always llcsnap."
        VARIATION     ipoaConfigPvcDefaultMtu
             ACCESS    read-only
             DESCRIPTION  "This implementation does not support
                           a set to this object."
        VARIATION     ipoaConfigPvcRowStatus
             DESCRIPTION  "This implementation does not support
                           this object."
    SUPPORTS             EtherLike-MIB    -- RFC 2665
        INCLUDES         { etherStatsBaseGroup,
                           etherDuplexGroup }
        VARIATION     dot3StatsTable
             DESCRIPTION  "The table entries are indexed by the
                           interface index of either, an interface
                           defined by the DEVICE profile statement,
                           or a dynamically generated OSA-Express
                           QDIO port interface."
        VARIATION     dot3StatsInternalMacTransmitErrors
             DESCRIPTION  "This implementation does not support
                           this object."
        VARIATION     dot3StatsFrameTooLongs
             DESCRIPTION  "This object is not supported for
                           OSA-Express QDIO Fast Ethernet adapters.
                           The object will be set to 0."
        VARIATION     dot3StatsInternalMacReceiveErrors
             DESCRIPTION  "This object is not supported for
                           OSA-Express QDIO Fast Ethernet adapters.
                           The object will be set to 0."
    ::= { ibmTcpIpMvsCaps 2 }
 ibmTcpIpMvsOspfCaps AGENT-CAPABILITIES
    PRODUCT-RELEASE  "IBM z/OS Communications Server
                      Version 2 Release 1 OSPF Subagent"
    STATUS           current
    DESCRIPTION      "IBM z/OS Communications Server
                      OSPF Subagent"
    SUPPORTS            OSPF-MIB   -- RFC 1850
        INCLUDES        { ospfBasicGroup,
                          ospfAreaGroup,
                          ospfStubAreaGroup,
                          ospfLsdbGroup,
                          ospfIfGroup,
                          ospfIfMetricGroup,
                          ospfVirtIfGroup,
                          ospfNbrGroup,
                          ospfVirtNbrGroup,
                          ospfExtLsdbGroup,
                          ospfAreaAggregateGroup }
        VARIATION     ospfRouterId
            ACCESS      read-only
            DESCRIPTION "Write access is not required, nor supported."
        VARIATION     ospfAdminStat
            SYNTAX      Status { enabled(1) }
            ACCESS      read-only
            DESCRIPTION "Write access is not required, nor supported.
                         This implementation always has at least one
```

```
                interface enabled."
VARIATION     ospfAdminStat
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION     ospfASBdrRtrStatus
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION     ospfTOSSupport
    SYNTAX      TruthValue { false(2) }
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                This implementation does not support
                type-of-service routing."
VARIATION     ospfStubTOS
    SYNTAX      TOSType ( 0 )
    DESCRIPTION "This implementation only supports TOS
                set to 0."
VARIATION     ospfExtLsdbLimit
    SYNTAX      Integer32 ( -1 )
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                This implementation does not have a limit
                on maximum number of non-default
                AS-external-LSAs entries."
VARIATION     ospfMulticastExtensions
    SYNTAX      Integer32 ( 0 )
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                This implementation does not support
                multicast forwarding."
VARIATION     ospfExitOverflowInterval
    ACCESS      not-implemented
    DESCRIPTION "This implementation does not support
                Overflow State."
VARIATION     ospfDemandExtensions
    SYNTAX      TruthValue { true(1) }
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                This router always supports demand routing."
VARIATION     ospfImportAsExtern
    SYNTAX      INTEGER { importExternal(1),
                          importNoExternal(2) }
    DESCRIPTION "This implementation only supports these
                import AS external link-state advertisement."
VARIATION     ospfImportAsExtern
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION     ospfAreaSummary
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION     ospfAreaStatus
    ACCESS      not-implemented
    DESCRIPTION "This implementation does not support
                this object."
VARIATION     ospfStubMetric
    ACCESS       read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION     ospfStubStatus
    ACCESS      not-implemented
    DESCRIPTION "This implementation does not support
                this object."
VARIATION     ospfStubMetricType
    SYNTAX      INTEGER { comparableCost(2),
                          nonComparable(3) }
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                This implementation only supports these
                types of metric advertised as a default
                route."
VARIATION     ospfLsdbType
    SYNTAX      INTEGER { routerLink(1), networklink(2),
                 summaryLink(3), asSummaryLink(4) }
    DESCRIPTION "This implementation only supports these
                types of links."
VARIATION     ospfAddressLessIf
    SYNTAX      Integer32 ( 0 )
    DESCRIPTION "This implementation only supports Interfaces
                with IP addresses."
VARIATION      ospfIfAreaId
```

```
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfType
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfAdminStat
    SYNTAX      Status { enabled(1) }
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                 This implementation only supports the value
                 formed on the interface, and the interface
                 will be advertised as an internal route
                 to some area."
VARIATION      ospfIfRtrPriority
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfTransitDelay
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfRetransInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfHelloInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfRtrDeadInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfPollInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfAuthKey
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfStatus
    ACCESS      not-implemented
    DESCRIPTION "This implementation does not support
                 this object."
VARIATION      ospfIfMulticastForwarding
    SYNTAX      INTEGER { blocked(1) }
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                 This implementation does not support
                 multicast forwarding."
VARIATION      ospfIfDemand
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfAuthType
    SYNTAX      INTEGER { none(0), simplePassword(1) }
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported.
                 This implementation only supports these
                 values."
VARIATION     ospfIfMetricAddressLessIf
    SYNTAX      Integer32 ( 0 )
    DESCRIPTION "This implementation only supports Interfaces
                 with IP addresses."
VARIATION      ospfIfMetricValue
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfIfMetricStatus
    ACCESS      not-implemented
    DESCRIPTION "This implementation does not support
                 this object."
VARIATION      ospfIfMetricTOS
    SYNTAX      TOSType ( 0 )
    DESCRIPTION "This implementation only supports value of 0."
VARIATION      ospfVirtIfTransitDelay
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfVirtIfRetransInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfVirtIfHelloInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
VARIATION      ospfVirtIfRtrDeadInterval
    ACCESS      read-only
    DESCRIPTION "Write access is not required, nor supported."
```

```
        VARIATION      ospfVirtIfAuthKey
            ACCESS      read-only
            DESCRIPTION "Write access is not required, nor supported."
        VARIATION      ospfVirtIfStatus
            ACCESS      not-implemented
            DESCRIPTION "This implementation does not support
                         this object."
        VARIATION      ospfVirtIfAuthType
            SYNTAX      INTEGER { none(0), simplePassword(1) }
            ACCESS      read-only
            DESCRIPTION "Write access is not required, nor supported.
                         This implementation only supports these
                         values."
        VARIATION     ospfNbrAddressLessIndex
            SYNTAX      InterfaceIndex ( 0 )
            DESCRIPTION "This implementation only supports Interfaces
                         with IP addresses."
        VARIATION      ospfNbrPriority
            ACCESS      read-only
            DESCRIPTION "Write access is not required, nor supported."
        VARIATION      ospfNbmaNbrStatus
            ACCESS      not-implemented
            DESCRIPTION "This implementation does not support
                         this object."
        VARIATION      ospfVirtNbrOptions
            SYNTAX      Integer32 ( 0 )
            DESCRIPTION "This implementation only supports value of 0."
        VARIATION      ospfAreaAggregateStatus
            ACCESS      not-implemented
            DESCRIPTION "This implementation does not support
                         this object."
        VARIATION      ospfAreaAggregateEffect
            ACCESS      read-only
            DESCRIPTION "Write access is not required, nor supported."
        VARIATION     ospfAreaAggregateLsdbType
            SYNTAX      INTEGER { summaryLink(3) }
            DESCRIPTION "This implementation only supports summary
                         link Lsdb Type."
    ::= { ibmTcpIpMvsCaps 3 }
ibmTcpIpMvsSlapm2Caps AGENT-CAPABILITIES
    PRODUCT-RELEASE  "IBM z/OS Communications Server
                      Version 2 Release 1 Network Service Level
                      Agreement subagent (nslapm2)"
    STATUS           current
    DESCRIPTION      "Network Service Level Agreement subagent"
    -- A copy of this MIB is installed as slapm2.mi2 in HFS at
    -- /usr/lpp/tcpip/samples as part of installing the
    -- IBM z/OS Communications Server.
    SUPPORTS             NETWORK-SLAPM2-MIB
        INCLUDES         { slapm2BaseGroup,
                           slapm2NotGroup }
        VARIATION   slapm2PolicyMonInterval
            SYNTAX  Unsigned32 (15..86400)
            DESCRIPTION
                                -- 15 second min, 24 hour max
                        "Only a minimum value of 30 seconds is
                         supported (30 second min, 24 hour max)."
        VARIATION   slapm2PRStatsInInProOctets
            DESCRIPTION  "Not supported. A value of zero is always
                          returned."
        VARIATION   slapm2PRStatsInInProPackets
            DESCRIPTION  "Not supported. A value of zero is always
                          returned."
     ::= { ibmTcpIpMvsCaps 5 }
 ibmMvsTN3270SaCaps  AGENT-CAPABILITIES
    PRODUCT-RELEASE  "IBM z/OS Communications Server
                      Version 2 Release 1 TN3270 subagent"
    STATUS           current
    DESCRIPTION      "TN3270 subagent"
    -- A copy of this MIB is installed as mvstn3270.mi2 in the HFS at
    -- /usr/lpp/tcpip/samples as part of installing the
    -- IBM z/OS Communications Server.
    SUPPORTS             IBMMVSTN3270-MIB
        INCLUDES         { ibmMvsTN3270ConnectionGroup,
                           ibmMvsTN3270MonitorGroup }
    ::= { ibmTcpIpMvsCaps 6 }
END
```

Appendix B. Management Information Base (MIB) objects

This topic lists the objects defined by the Management Information Base (MIB), which are supported by the SNMP agent and subagents on the z/OS Communications Server, and the maximum access allowed.

Note: If an SNMP SET (write) is attempted against a variable for which the maximum access is read-only, an error code is returned. For an SNMPv2 request, the error code is noAccess or notWritable.

The object types are defined using the following fields:

Object Descriptor
A textual name for the object type, along with its corresponding OBJECT IDENTIFIER.

Object Identifier
The name for the object type, using ASN.1 notation.

Supported by
Support by the agent or subagents. If support is by one of the subagents, the subagent is named. Supported subagents include:

- TCP/IP
- OMPRoute
- Network SLAPM2
- TN3270

Defined by
The location of the description of the object.

The SNMP agent provides support of the following Enterprise-specific MIBs:

- Subagent MIB
- Extensions to the DPI20 MIB defined by RFC 1592

The TCP/IP subagent provides support of the following Enterprise-specific MIBs:

- IBM 3172 MIB
- IBM TCP/IP MVS Enterprise-specific MIB (which includes Remote Ping)

The TN3270 subagent provides support of the TN3270 Enterprise-specific MIB.

Copies of the SMI syntax for the previously mentioned MIBs are installed in the z/OS UNIX file system directory /usr/lpp/tcpip/samples as:

- mvstcpip.mi2 (SMIv2)
- saMIB.mi2 (SMIv2)
- saMIB.mib (SMIv1)
- slapm2.mi2 (SMIv2)
- rfc1592b.mi2 (SMIv2)
- rfc1592b.mib (SMIv1)
- ibm3172.mi2 (SMIv2)
- ibm3172.mib (SMIv1)

- mvstn3270.mi2 (SMIv2)

Access Allowed

- Read-only (R/O)
- Read-write (R/W)
- Read-create (R/C)
- Write-only (W/O)
- Not-accessible (N/A)

Table 22 on page 1027 shows the MIB objects supported by z/OS Communications Server IP SNMP agent and subagents.

Table 22. MIB objects

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
sysDescr	1.3.6.1.2.1.1.1	Agent	RFC1907	R/O
sysObjectID	1.3.6.1.2.1.1.2	Agent	RFC1907	R/O
sysUpTime	1.3.6.1.2.1.1.3	Agent	RFC1907	R/O
sysContact	1.3.6.1.2.1.1.4	Agent	RFC1907	R/W
sysName	1.3.6.1.2.1.1.5	Agent	RFC1907	R/W
sysLocation	1.3.6.1.2.1.1.6	Agent	RFC1907	R/W
sysServices	1.3.6.1.2.1.1.7	Agent	RFC1907	R/O
sysORLastChange	1.3.6.1.2.1.1.8	Agent	RFC1907	R/O
sysORTable	1.3.6.1.2.1.1.9	Agent	RFC1907	N/A
sysOREntry	1.3.6.1.2.1.1.9.1	Agent	RFC1907	N/A
sysORIndex	1.3.6.1.2.1.1.9.1.1	Agent	RFC1907	N/A
sysORID	1.3.6.1.2.1.1.9.1.2	Agent	RFC1907	R/O
sysORDescr	1.3.6.1.2.1.1.9.1.3	Agent	RFC1907	R/O
sysORUpTime	1.3.6.1.2.1.1.9.1.4	Agent	RFC1907	R/O
ifNumber	1.3.6.1.2.1.2.1	TCP/IP	RFC2233	R/O
ifTable	1.3.6.1.2.1.2.2	TCP/IP	RFC2233	N/A
ifEntry	1.3.6.1.2.1.2.2.1	TCP/IP	RFC2233	N/A
ifIndex	1.3.6.1.2.1.2.2.1.1	TCP/IP	RFC2233	R/O
ifDescr	1.3.6.1.2.1.2.2.1.2	TCP/IP	RFC2233	R/O
ifType	1.3.6.1.2.1.2.2.1.3	TCP/IP	RFC2233	R/O
ifMtu	1.3.6.1.2.1.2.2.1.4	TCP/IP	RFC2233	R/O
ifSpeed	1.3.6.1.2.1.2.2.1.5	TCP/IP	RFC2233	R/O
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	TCP/IP	RFC2233	R/O
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	TCP/IP	RFC2233	R/W
ifOperStatus	1.3.6.1.2.1.2.2.1.8	TCP/IP	RFC2233	R/O
ifLastChange	1.3.6.1.2.1.2.2.1.9	TCP/IP	RFC2233	R/O
ifInOctets	1.3.6.1.2.1.2.2.1.10	TCP/IP	RFC2233	R/O
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	TCP/IP	RFC2233	R/O
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	TCP/IP	RFC2233	R/O
ifInDiscards	1.3.6.1.2.1.2.2.1.13	TCP/IP	RFC2233	R/O
ifInErrors	1.3.6.1.2.1.2.2.1.14	TCP/IP	RFC2233	R/O
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	TCP/IP	RFC2233	R/O
ifOutOctets	1.3.6.1.2.1.2.2.1.16	TCP/IP	RFC2233	R/O
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	TCP/IP	RFC2233	R/O
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	TCP/IP	RFC2233	R/O
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	TCP/IP	RFC2233	R/O
ifOutErrors	1.3.6.1.2.1.2.2.1.20	TCP/IP	RFC2233	R/O
ifOutQLen	1.3.6.1.2.1.2.2.1.21	TCP/IP	RFC2233	R/O
ifSpecific	1.3.6.1.2.1.2.2.1.22	TCP/IP	RFC2233	R/O
ipForwarding	1.3.6.1.2.1.4.1	TCP/IP	RFC4293	R/W
ipDefaultTTL	1.3.6.1.2.1.4.2	TCP/IP	RFC4293	R/W

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipInReceives	1.3.6.1.2.1.4.3	TCP/IP	RFC4293	R/O
ipInHdrErrors	1.3.6.1.2.1.4.4	TCP/IP	RFC4293	R/O
ipInAddrErrors	1.3.6.1.2.1.4.5	TCP/IP	RFC4293	R/O
ipForwDatagrams	1.3.6.1.2.1.4.6	TCP/IP	RFC4293	R/O
ipInUnknownProtos	1.3.6.1.2.1.4.7	TCP/IP	RFC4293	R/O
ipInDiscards	1.3.6.1.2.1.4.8	TCP/IP	RFC4293	R/O
ipInDelivers	1.3.6.1.2.1.4.9	TCP/IP	RFC4293	R/O
ipOutRequests	1.3.6.1.2.1.4.10	TCP/IP	RFC4293	R/O
ipOutDiscards	1.3.6.1.2.1.4.11	TCP/IP	RFC4293	R/O
ipOutNoRoutes	1.3.6.1.2.1.4.12	TCP/IP	RFC4293	R/O
ipReasmTimeout	1.3.6.1.2.1.4.13	TCP/IP	RFC4293	R/W
ipReasmReqds	1.3.6.1.2.1.4.14	TCP/IP	RFC4293	R/O
ipReasmOKs	1.3.6.1.2.1.4.15	TCP/IP	RFC4293	R/O
ipReasmFails	1.3.6.1.2.1.4.16	TCP/IP	RFC4293	R/O
ipFragOKs	1.3.6.1.2.1.4.17	TCP/IP	RFC4293	R/O
ipFragFails	1.3.6.1.2.1.4.18	TCP/IP	RFC4293	R/O
ipFragCreates	1.3.6.1.2.1.4.19	TCP/IP	RFC4293	R/O
ipAddrTable	1.3.6.1.2.1.4.20	TCP/IP	RFC4293	N/A
ipAddrEntry	1.3.6.1.2.1.4.20.1	TCP/IP	RFC4293	N/A
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	TCP/IP	RFC4293	R/O
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	TCP/IP	RFC4293	R/O
ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	TCP/IP	RFC4293	R/O
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	TCP/IP	RFC4293	R/O
ipAdEntReasmMaxSize	1.3.6.1.2.1.4.20.1.5	TCP/IP	RFC4293	R/O
ipNetToMediaTable	1.3.6.1.2.1.4.22	TCP/IP	RFC4293	N/A
ipNetToMediaEntry	1.3.6.1.2.1.4.22.1	TCP/IP	RFC4293	N/A
ipNetToMediaIfIndex	1.3.6.1.2.1.4.22.1.1	TCP/IP	RFC4293	R/O
ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2	TCP/IP	RFC4293	R/O
ipNetToMediaNetAddress	1.3.6.1.2.1.4.22.1.3	TCP/IP	RFC4293	R/O
ipNetToMediaType	1.3.6.1.2.1.4.22.1.4	TCP/IP	RFC4293	R/O
ipRoutingDiscards	1.3.6.1.2.1.4.23	TCP/IP	RFC4293	R/O
ipForward	1.3.6.1.2.1.4.24	TCP/IP	RFC1354	N/A
ipForwardNumber	1.3.6.1.2.1.4.24.1	TCP/IP	RFC1354	R/O
ipForwardTable	1.3.6.1.2.1.4.24.2	TCP/IP	RFC1354	N/A
ipForwardEntry	1.3.6.1.2.1.4.24.2.1	TCP/IP	RFC1354	N/A
ipForwardDest	1.3.6.1.2.1.4.24.2.1.1	TCP/IP	RFC1354	R/O
ipForwardMask	1.3.6.1.2.1.4.24.2.1.2	TCP/IP	RFC1354	R/O
ipForwardPolicy	1.3.6.1.2.1.4.24.2.1.3	TCP/IP	RFC1354	R/O
ipForwardNextHop	1.3.6.1.2.1.4.24.2.1.4	TCP/IP	RFC1354	R/O
ipForwardIfIndex	1.3.6.1.2.1.4.24.2.1.5	TCP/IP	RFC1354	R/O
ipForwardType	1.3.6.1.2.1.4.24.2.1.6	TCP/IP	RFC1354	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipForwardProto	1.3.6.1.2.1.4.24.2.1.7	TCP/IP	RFC1354	R/O
ipForwardAge	1.3.6.1.2.1.4.24.2.1.8	TCP/IP	RFC1354	R/O
ipForwardInfo	1.3.6.1.2.1.4.24.2.1.9	TCP/IP	RFC1354	R/O
ipForwardNextHopAS	1.3.6.1.2.1.4.24.2.1.10	TCP/IP	RFC1354	R/O
ipForwardMetric1	1.3.6.1.2.1.4.24.2.1.11	TCP/IP	RFC1354	R/O
ipForwardMetric2	1.3.6.1.2.1.4.24.2.1.12	TCP/IP	RFC1354	R/O
ipForwardMetric3	1.3.6.1.2.1.4.24.2.1.13	TCP/IP	RFC1354	R/O
ipForwardMetric4	1.3.6.1.2.1.4.24.2.1.14	TCP/IP	RFC1354	R/O
ipForwardMetric5	1.3.6.1.2.1.4.24.2.1.15	TCP/IP	RFC1354	R/O
inetCidrRouteNumber	1.3.6.1.2.1.4.24.6	TCP/IP	RFC4292	R/O
inetCidrRouteTable	1.3.6.1.2.1.4.24.7	TCP/IP	RFC4292	N/A
inetCidrRouteEntry	1.3.6.1.2.1.4.24.7.1	TCP/IP	RFC4292	N/A
inetCidrRouteDestType	1.3.6.1.2.1.4.24.7.1.1	TCP/IP	RFC4292	N/A
inetCidrRouteDest	1.3.6.1.2.1.4.24.7.1.2	TCP/IP	RFC4292	N/A
inetCidrRoutePfxLen	1.3.6.1.2.1.4.24.7.1.3	TCP/IP	RFC4292	N/A
inetCidrRoutePolicy	1.3.6.1.2.1.4.24.7.1.4	TCP/IP	RFC4292	N/A
inetCidrRouteNextHopType	1.3.6.1.2.1.4.24.7.1.5	TCP/IP	RFC4292	N/A
inetCidrRouteNextHop	1.3.6.1.2.1.4.24.7.1.6	TCP/IP	RFC4292	N/A
inetCidrRouteIfIndex	1.3.6.1.2.1.4.24.7.1.7	TCP/IP	RFC4292	R/O
inetCidrRouteType	1.3.6.1.2.1.4.24.7.1.8	TCP/IP	RFC4292	R/O
inetCidrRouteProto	1.3.6.1.2.1.4.24.7.1.9	TCP/IP	RFC4292	R/O
inetCidrRouteAge	1.3.6.1.2.1.4.24.7.1.10	TCP/IP	RFC4292	R/O
inetCidrRouteNextHopAS	1.3.6.1.2.1.4.24.7.1.11	TCP/IP	RFC4292	R/O
inetCidrRouteMetric1	1.3.6.1.2.1.4.24.7.1.12	TCP/IP	RFC4292	R/O
inetCidrRouteMetric2	1.3.6.1.2.1.4.24.7.1.13	TCP/IP	RFC4292	R/O
inetCidrRouteMetric3	1.3.6.1.2.1.4.24.7.1.14	TCP/IP	RFC4292	R/O
inetCidrRouteMetric4	1.3.6.1.2.1.4.24.7.1.15	TCP/IP	RFC4292	R/O
inetCidrRouteMetric5	1.3.6.1.2.1.4.24.7.1.16	TCP/IP	RFC4292	R/O
inetCidrRouteStatus	1.3.6.1.2.1.4.24.7.1.17	TCP/IP	RFC4292	R/O
inetCidrRouteDiscards	1.3.6.1.2.1.4.24.8	TCP/IP	RFC4292	R/O
ipv6IpForwarding	1.3.6.1.2.1.4.25	TCP/IP	RFC4293	R/W
ipv6IpDefaultHopLimit	1.3.6.1.2.1.4.26	TCP/IP	RFC4293	R/W
ipv6IfTableLastChange	1.3.6.1.2.1.4.29	TCP/IP	RFC4293	R/O
ipv6InterfaceTable	1.3.6.1.2.1.4.30	TCP/IP	RFC4293	N/A
ipv6InterfaceEntry	1.3.6.1.2.1.4.30.1	TCP/IP	RFC4293	N/A
ipv6InterfaceIfIndex	1.3.6.1.2.1.4.30.1.1	TCP/IP	RFC4293	N/A
ipv6InterfaceReasmMaxSize	1.3.6.1.2.1.4.30.1.2	TCP/IP	RFC4293	R/O
ipv6InterfaceIdentifier	1.3.6.1.2.1.4.30.1.3	TCP/IP	RFC4293	R/O
ipv6interfaceEnableStatus	1.3.6.1.2.1.4.30.1.4	TCP/IP	RFC4293	R/O
ipv6InterfaceReachableTime	1.3.6.1.2.1.4.30.1.5	TCP/IP	RFC4293	R/O
ipv6InterfaceRetransmitTime	1.3.6.1.2.1.4.30.1.6	TCP/IP	RFC4293	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipv6InterfaceForwarding	1.3.6.1.2.1.4.30.1.7	TCP/IP	RFC4293	R/O
ipTrafficStats	1.3.6.1.2.1.4.31	TCP/IP	RFC4293	N/A
ipSystemStatsTable	1.3.6.1.2.1.4.31.1	TCP/IP	RFC4293	N/A
ipSystemStatsEntry	1.3.6.1.2.1.4.31.1.1	TCP/IP	RFC4293	N/A
ipSystemStatsAFType	1.3.6.1.2.1.4.31.1.1.1	TCP/IP	RFC4293	N/A
ipSystemStatsInReceives	1.3.6.1.2.1.4.31.1.1.3	TCP/IP	RFC4293	R/O
ipSystemStatsHCInReceives	1.3.6.1.2.1.4.31.1.1.4	TCP/IP	RFC4293	R/O
ipSystemStatsInOctets	1.3.6.1.2.1.4.31.1.1.5	TCP/IP	RFC4293	R/O
ipSystemStatsHCInOctets	1.3.6.1.2.1.4.31.1.1.6	TCP/IP	RFC4293	R/O
ipSystemStatsInHdrErrors	1.3.6.1.2.1.4.31.1.1.7	TCP/IP	RFC4293	R/O
ipSystemStatsInNoRoutes	1.3.6.1.2.1.4.31.1.1.8	TCP/IP	RFC4293	R/O
ipSystemStatsInAddrErrors	1.3.6.1.2.1.4.31.1.1.9	TCP/IP	RFC4293	R/O
ipSystemStatsInUnknownProtos	1.3.6.1.2.1.4.31.1.1.10	TCP/IP	RFC4293	R/O
ipSystemStatsInTruncatedPkts	1.3.6.1.2.1.4.31.1.1.11	TCP/IP	RFC4293	R/O
ipSystemStatsInForwDatagrams	1.3.6.1.2.1.4.31.1.1.12	TCP/IP	RFC4293	R/O
ipSystemStatsHCInForwDatagrams	1.3.6.1.2.1.4.31.1.1.13	TCP/IP	RFC4293	R/O
ipSystemStatsReasmReqds	1.3.6.1.2.1.4.31.1.1.14	TCP/IP	RFC4293	R/O
ipSystemStatsReasmOKs	1.3.6.1.2.1.4.31.1.1.15	TCP/IP	RFC4293	R/O
ipSystemStatsReasmFails	1.3.6.1.2.1.4.31.1.1.16	TCP/IP	RFC4293	R/O
ipSystemStatsInDiscards	1.3.6.1.2.1.4.31.1.1.17	TCP/IP	RFC4293	R/O
ipSystemStatsInDelivers	1.3.6.1.2.1.4.31.1.1.18	TCP/IP	RFC4293	R/O
ipSystemStatsHCInDelivers	1.3.6.1.2.1.4.31.1.1.19	TCP/IP	RFC4293	R/O
ipSystemStatsOutRequests	1.3.6.1.2.1.4.31.1.1.20	TCP/IP	RFC4293	R/O
ipSystemStatsHCOutRequests	1.3.6.1.2.1.4.31.1.1.21	TCP/IP	RFC4293	R/O
ipSystemStatsOutNoRoutes	1.3.6.1.2.1.4.31.1.1.22	TCP/IP	RFC4293	R/O
ipSystemStatsOutForwDatagrams	1.3.6.1.2.1.4.31.1.1.23	TCP/IP	RFC4293	R/O
ipSystemStatsHCOutForwDatagrams	1.3.6.1.2.1.4.31.1.1.24	TCP/IP	RFC4293	R/O
ipSystemStatsOutDiscards	1.3.6.1.2.1.4.31.1.1.25	TCP/IP	RFC4293	R/O
ipSystemStatsOutFragReqds	1.3.6.1.2.1.4.31.1.1.26	TCP/IP	RFC4293	R/O
ipSystemStatsOutFragOKs	1.3.6.1.2.1.4.31.1.1.27	TCP/IP	RFC4293	R/O
ipSystemStatsOutFragFails	1.3.6.1.2.1.4.31.1.1.28	TCP/IP	RFC4293	R/O
ipSystemStatsOutFragCreates	1.3.6.1.2.1.4.31.1.1.29	TCP/IP	RFC4293	R/O
ipSystemStatsOutTransmits	1.3.6.1.2.1.4.31.1.1.30	TCP/IP	RFC4293	R/O
ipSystemStatsHCOutTransmits	1.3.6.1.2.1.4.31.1.1.31	TCP/IP	RFC4293	R/O
ipSystemStatsOutOctets	1.3.6.1.2.1.4.31.1.1.32	TCP/IP	RFC4293	R/O
ipSystemStatsHCOutOctets	1.3.6.1.2.1.4.31.1.1.33	TCP/IP	RFC4293	R/O
ipSystemStatsInMcastPkts	1.3.6.1.2.1.4.31.1.1.34	TCP/IP	RFC4293	R/O
ipSystemStatsHCInMcastPkts	1.3.6.1.2.1.4.31.1.1.35	TCP/IP	RFC4293	R/O
ipSystemStatsInMcastOctets	1.3.6.1.2.1.4.31.1.1.36	TCP/IP	RFC4293	R/O
ipSystemStatsHCInMcastOctets	1.3.6.1.2.1.4.31.1.1.37	TCP/IP	RFC4293	R/O
ipSystemStatsOutMcastPkts	1.3.6.1.2.1.4.31.1.1.38	TCP/IP	RFC4293	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipSystemStatsHCOutMcastPkts	1.3.6.1.2.1.4.31.1.1.39	TCP/IP	RFC4293	R/O
ipSystemStatsOutMcastOctets	1.3.6.1.2.1.4.31.1.1.40	TCP/IP	RFC4293	R/O
ipSystemStatsHCOutMcastOctets	1.3.6.1.2.1.4.31.1.1.41	TCP/IP	RFC4293	R/O
ipSystemStatsInBcastPkts	1.3.6.1.2.1.4.31.1.1.42	TCP/IP	RFC4293	R/O
ipSystemStatsHCInBcastPkts	1.3.6.1.2.1.4.31.1.1.43	TCP/IP	RFC4293	R/O
ipSystemStatsOutBcastPkts	1.3.6.1.2.1.4.31.1.1.44	TCP/IP	RFC4293	R/O
ipSystemStatsHCOutBcastPkts	1.3.6.1.2.1.4.31.1.1.45	TCP/IP	RFC4293	R/O
ipSystemStatsDiscontinuityTime	1.3.6.1.2.1.4.31.1.1.46	TCP/IP	RFC4293	R/O
ipSystemStatsRefreshRate	1.3.6.1.2.1.4.31.1.1.47	TCP/IP	RFC4293	R/O
ipAddressPrefixTable	1.3.6.1.2.1.4.32	TCP/IP	RFC4293	N/A
ipAddressPrefixEntry	1.3.6.1.2.1.4.32.1	TCP/IP	RFC4293	N/A
ipAddressPrefixIfIndex	1.3.6.1.2.1.4.32.1.1	TCP/IP	RFC4293	N/A
ipAddressPrefixType	1.3.6.1.2.1.4.32.1.2	TCP/IP	RFC4293	N/A
ipAddressPrefixPrefix	1.3.6.1.2.1.4.32.1.3	TCP/IP	RFC4293	N/A
ipAddressPrefixLength	1.3.6.1.2.1.4.32.1.4	TCP/IP	RFC4293	N/A
ipAddressPrefixOrigin	1.3.6.1.2.1.4.32.1.5	TCP/IP	RFC4293	R/O
ipAddressPrefixOnLinkFlag	1.3.6.1.2.1.4.32.1.6	TCP/IP	RFC4293	R/O
ipAddressPrefixAutonomousFlag	1.3.6.1.2.1.4.32.1.7	TCP/IP	RFC4293	R/O
ipAddressPrefixAdvPreferredLifetime	1.3.6.1.2.1.4.32.1.8	TCP/IP	RFC4293	R/O
ipAddressPrefixAdvValidLifetime	1.3.6.1.2.1.4.32.1.9	TCP/IP	RFC4293	R/O
ipAddressSpinLock	1.3.6.1.2.1.4.33	TCP/IP	RFC4293	N/A
ipAddressTable	1.3.6.1.2.1.4.34	TCP/IP	RFC4293	N/A
ipAddressEntry	1.3.6.1.2.1.4.34.1	TCP/IP	RFC4293	N/A
ipAddressAddrType	1.3.6.1.2.1.4.34.1.1	TCP/IP	RFC4293	N/A
ipAddressAddr	1.3.6.1.2.1.4.34.1.2	TCP/IP	RFC4293	N/A
ipAddressIfIndex	1.3.6.1.2.1.4.34.1.3	TCP/IP	RFC4293	N/A
ipAddressType	1.3.6.1.2.1.4.34.1.4	TCP/IP	RFC4293	R/O
ipAddressPrefix	1.3.6.1.2.1.4.34.1.5	TCP/IP	RFC4293	R/O
ipAddressOrigin	1.3.6.1.2.1.4.34.1.6	TCP/IP	RFC4293	R/O
ipAddressStatus	1.3.6.1.2.1.4.34.1.7	TCP/IP	RFC4293	R/O
ipAddressCreated	1.3.6.1.2.1.4.34.1.8	TCP/IP	RFC4293	R/O
ipAddressLastChanged	1.3.6.1.2.1.4.34.1.9	TCP/IP	RFC4293	R/O
ipAddressRowStatus	1.3.6.1.2.1.4.34.1.10	TCP/IP	RFC4293	R/O
ipAddressStorageType	1.3.6.1.2.1.4.34.1.11	TCP/IP	RFC4293	R/O
ipNetToPhysicalTable	1.3.6.1.2.1.4.35	TCP/IP	RFC4293	R/O
ipNetToPhysicalEntry	1.3.6.1.2.1.4.35.1	TCP/IP	RFC4293	R/O
ipNetToPhysicalIfIndex	1.3.6.1.2.1.4.35.1.1	TCP/IP	RFC4293	R/O
ipNetToPhysicalNetAddressType	1.3.6.1.2.1.4.35.1.2	TCP/IP	RFC4293	R/O
ipNetToPhysicalNetAddress	1.3.6.1.2.1.4.35.1.3	TCP/IP	RFC4293	R/O
ipNetToPhysicalPhysAddress	1.3.6.1.2.1.4.35.1.4	TCP/IP	RFC4293	R/O
ipNetToPhysicalLastUpdated	1.3.6.1.2.1.4.35.1.5	TCP/IP	RFC4293	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipNetToPhysicalType	1.3.6.1.2.1.4.35.1.6	TCP/IP	RFC4293	R/O
ipNetToPhysicalState	1.3.6.1.2.1.4.35.1.7	TCP/IP	RFC4293	R/O
ipNetToPhysicalRowStatus	1.3.6.1.2.1.4.35.1.8	TCP/IP	RFC4293	R/O
ipDefaultRouterTable	1.3.6.1.2.1.4.37	TCP/IP	RFC4293	N/A
ipDefaultRouterEntry	1.3.6.1.2.1.4.37.1	TCP/IP	RFC4293	N/A
ipDefaultRouterAFType	1.3.6.1.2.1.4.37.1.1	TCP/IP	RFC4293	N/A
ipDefaultRouterAddress	1.3.6.1.2.1.4.37.1.2	TCP/IP	RFC4293	N/A
ipDefaultRouterIfIndex	1.3.6.1.2.1.4.37.1.3	TCP/IP	RFC4293	N/A
ipDefaultRouterLifetime	1.3.6.1.2.1.4.37.1.4	TCP/IP	RFC4293	R/O
ipDefaultRouterPreference	1.3.6.1.2.1.4.37.1.5	TCP/IP	RFC4293	R/O
icmpInMsgs	1.3.6.1.2.1.5.1	TCP/IP	RFC4293	R/O
icmpInErrors	1.3.6.1.2.1.5.2	TCP/IP	RFC4293	R/O
icmpInDestUnreachs	1.3.6.1.2.1.5.3	TCP/IP	RFC4293	R/O
icmpInTimeExcds	1.3.6.1.2.1.5.4	TCP/IP	RFC4293	R/O
icmpInParmProbs	1.3.6.1.2.1.5.5	TCP/IP	RFC4293	R/O
icmpInSrcQuenchs	1.3.6.1.2.1.5.6	TCP/IP	RFC4293	R/O
icmpInRedirects	1.3.6.1.2.1.5.7	TCP/IP	RFC4293	R/O
icmpInEchos	1.3.6.1.2.1.5.8	TCP/IP	RFC4293	R/O
icmpInEchoReps	1.3.6.1.2.1.5.9	TCP/IP	RFC4293	R/O
icmpInTimestamps	1.3.6.1.2.1.5.10	TCP/IP	RFC4293	R/O
icmpInTimestampReps	1.3.6.1.2.1.5.11	TCP/IP	RFC4293	R/O
icmpInAddrMasks	1.3.6.1.2.1.5.12	TCP/IP	RFC4293	R/O
icmpInAddrMaskReps	1.3.6.1.2.1.5.13	TCP/IP	RFC4293	R/O
icmpOutMsgs	1.3.6.1.2.1.5.14	TCP/IP	RFC4293	R/O
icmpOutErrors	1.3.6.1.2.1.5.15	TCP/IP	RFC4293	R/O
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	TCP/IP	RFC4293	R/O
icmpOutTimeExcds	1.3.6.1.2.1.5.17	TCP/IP	RFC4293	R/O
icmpOutParmProbs	1.3.6.1.2.1.5.18	TCP/IP	RFC4293	R/O
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	TCP/IP	RFC4293	R/O
icmpOutRedirects	1.3.6.1.2.1.5.20	TCP/IP	RFC4293	R/O
icmpOutEchos	1.3.6.1.2.1.5.21	TCP/IP	RFC4293	R/O
icmpOutEchoReps	1.3.6.1.2.1.5.22	TCP/IP	RFC4293	R/O
icmpOutTimestamps	1.3.6.1.2.1.5.23	TCP/IP	RFC4293	R/O
icmpOutTimestampReps	1.3.6.1.2.1.5.24	TCP/IP	RFC4293	R/O
icmpOutAddrMasks	1.3.6.1.2.1.5.25	TCP/IP	RFC4293	R/O
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	TCP/IP	RFC4293	R/O
icmpStatsTable	1.3.6.1.2.1.5.29	TCP/IP	RFC4293	N/A
icmpStatsEntry	1.3.6.1.2.1.5.29.1	TCP/IP	RFC4293	N/A
icmpStatsIPVersion	1.3.6.1.2.1.5.29.1.1	TCP/IP	RFC4293	N/A
icmpStatsInMsgs	1.3.6.1.2.1.5.29.1.2	TCP/IP	RFC4293	R/O
icmpStatsInErrors	1.3.6.1.2.1.5.29.1.3	TCP/IP	RFC4293	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
icmpStatsOutMsgs	1.3.6.1.2.1.5.29.1.4	TCP/IP	RFC4293	R/O
icmpStatsOutErrors	1.3.6.1.2.1.5.29.1.5	TCP/IP	RFC4293	R/O
icmpMsgStatsTable	1.3.6.1.2.1.5.30	TCP/IP	RFC4293	N/A
icmpMsgStatsEntry	1.3.6.1.2.1.5.30.1	TCP/IP	RFC4293	N/A
icmpMsgStatsIPVersion	1.3.6.1.2.1.5.30.1.1	TCP/IP	RFC4293	N/A
icmpMsgStatsType	1.3.6.1.2.1.5.30.1.2	TCP/IP	RFC4293	N/A
icmpMsgStatsInPkts	1.3.6.1.2.1.5.30.1.3	TCP/IP	RFC4293	R/O
icmpMsgStatsOutPkts	1.3.6.1.2.1.5.30.1.4	TCP/IP	RFC4293	R/O
tcpRtoAlgorithm	1.3.6.1.2.1.6.1	TCP/IP	RFC4022	R/O
tcpRtoMin	1.3.6.1.2.1.6.2	TCP/IP	RFC4022	R/O
tcpRtoMax	1.3.6.1.2.1.6.3	TCP/IP	RFC4022	R/O
tcpMaxConn	1.3.6.1.2.1.6.4	TCP/IP	RFC4022	R/O
tcpActiveOpens	1.3.6.1.2.1.6.5	TCP/IP	RFC4022	R/O
tcpPassiveOpens	1.3.6.1.2.1.6.6	TCP/IP	RFC4022	R/O
tcpAttemptFails	1.3.6.1.2.1.6.7	TCP/IP	RFC4022	R/O
tcpEstabResets	1.3.6.1.2.1.6.8	TCP/IP	RFC4022	R/O
tcpCurrEstab	1.3.6.1.2.1.6.9	TCP/IP	RFC4022	R/O
tcpInSegs	1.3.6.1.2.1.6.10	TCP/IP	RFC4022	R/O
tcpOutSegs	1.3.6.1.2.1.6.11	TCP/IP	RFC4022	R/O
tcpRetransSegs	1.3.6.1.2.1.6.12	TCP/IP	RFC4022	R/O
tcpConnTable	1.3.6.1.2.1.6.13	TCP/IP	RFC4022	N/A
tcpConnEntry	1.3.6.1.2.1.6.13.1	TCP/IP	RFC4022	N/A
tcpConnState	1.3.6.1.2.1.6.13.1.1	TCP/IP	RFC4022	R/W
tcpConnLocalAddress	1.3.6.1.2.1.6.13.1.2	TCP/IP	RFC4022	R/O
tcpConnLocalPort	1.3.6.1.2.1.6.13.1.3	TCP/IP	RFC4022	R/O
tcpConnRemAddress	1.3.6.1.2.1.6.13.1.4	TCP/IP	RFC4022	R/O
tcpConnRemPort	1.3.6.1.2.1.6.13.1.5	TCP/IP	RFC4022	R/O
tcpInErrs	1.3.6.1.2.1.6.14	TCP/IP	RFC4022	R/O
tcpOutRsts	1.3.6.1.2.1.6.15	TCP/IP	RFC4022	R/O
tcpHCInSegs	1.3.6.1.2.1.6.17	TCP/IP	RFC4022	R/O
tcpHCOutSegs	1.3.6.1.2.1.6.18	TCP/IP	RFC4022	R/O
tcpConnectionTable	1.3.6.1.2.1.6.19	TCP/IP	RFC4022	N/A
tcpConnectionEntry	1.3.6.1.2.1.6.19.1	TCP/IP	RFC4022	N/A
tcpConnectionLocalAddressType	1.3.6.1.2.1.6.19.1.1	TCP/IP	RFC4022	N/A
tcpConnectionLocalAddress	1.3.6.1.2.1.6.19.1.2	TCP/IP	RFC4022	N/A
tcpConnectionLocalPort	1.3.6.1.2.1.6.19.1.3	TCP/IP	RFC4022	N/A
tcpConnectionRemAddressType	1.3.6.1.2.1.6.19.1.4	TCP/IP	RFC4022	N/A
tcpConnectionRemAddress	1.3.6.1.2.1.6.19.1.5	TCP/IP	RFC4022	N/A
tcpConnectionRemPort	1.3.6.1.2.1.6.19.1.6	TCP/IP	RFC4022	N/A
tcpConnectionState	1.3.6.1.2.1.6.19.1.7	TCP/IP	RFC4022	R/W
tcpConnectionProcess	1.3.6.1.2.1.6.19.1.8	TCP/IP	RFC4022	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
tcpListenerTable	1.3.6.1.2.1.6.20	TCP/IP	RFC4022	N/A
tcpListenerEntry	1.3.6.1.2.1.6.20.1	TCP/IP	RFC4022	N/A
tcpListenerLocalAddressType	1.3.6.1.2.1.6.20.1.1	TCP/IP	RFC4022	N/A
tcpListenerLocalAddress	1.3.6.1.2.1.6.20.1.2	TCP/IP	RFC4022	N/A
tcpListenerLocalPort	1.3.6.1.2.1.6.20.1.3	TCP/IP	RFC4022	N/A
tcpListenerProcess	1.3.6.1.2.1.6.20.1.4	TCP/IP	RFC4022	R/O
udpInDatagrams	1.3.6.1.2.1.7.1	TCP/IP	RFC4113	R/O
udpNoPorts	1.3.6.1.2.1.7.2	TCP/IP	RFC4113	R/O
udpInErrors	1.3.6.1.2.1.7.3	TCP/IP	RFC4113	R/O
udpOutDatagrams	1.3.6.1.2.1.7.4	TCP/IP	RFC4113	R/O
udpTable	1.3.6.1.2.1.7.5	TCP/IP	RFC4113	N/A
udpEntry	1.3.6.1.2.1.7.5.1	TCP/IP	RFC4113	N/A
udpLocalAddress	1.3.6.1.2.1.7.5.1.1	TCP/IP	RFC4113	R/O
udpLocalPort	1.3.6.1.2.1.7.5.1.2	TCP/IP	RFC4113	R/O
udpEndpointTable	1.3.6.1.2.1.7.7	TCP/IP	RFC4113	N/A
udpEndpointEntry	1.3.6.1.2.1.7.7.1	TCP/IP	RFC4113	N/A
udpEndpointLocalAddressType	1.3.6.1.2.1.7.7.1.1	TCP/IP	RFC4113	N/A
udpEndpointLocalAddress	1.3.6.1.2.1.7.7.1.2	TCP/IP	RFC4113	N/A
udpEndpointLocalPort	1.3.6.1.2.1.7.7.1.3	TCP/IP	RFC4113	N/A
udpEndpointRemoteAddressType	1.3.6.1.2.1.7.7.1.4	TCP/IP	RFC4113	N/A
udpEndpointRemoteAddress	1.3.6.1.2.1.7.7.1.5	TCP/IP	RFC4113	N/A
udpEndpointRemotePort	1.3.6.1.2.1.7.7.1.6	TCP/IP	RFC4113	N/A
udpEndpointInstance	1.3.6.1.2.1.7.7.1.7	TCP/IP	RFC4113	R/O
udpEndpointProcess	1.3.6.1.2.1.7.7.1.8	TCP/IP	RFC4113	R/O
udpHCInDatagrams	1.3.6.1.2.1.7.8	TCP/IP	RFC4113	R/O
udpHCOutDatagrams	1.3.6.1.2.1.7.9	TCP/IP	RFC4113	R/O
dot3StatsTable	1.3.6.1.2.1.10.7.2	TCP/IP	RCF2665	N/A
dot3StatsEntry	1.3.6.1.2.1.10.7.2.1	TCP/IP	RCF2665	N/A
dot3StatsIndex	1.3.6.1.2.1.10.7.2.1.1	TCP/IP	RCF2665	R/O
dot3StatsAlignmentErrors	1.3.6.1.2.1.10.7.2.1.2	TCP/IP	RCF2665	R/O
dot3StatsFCSErrors	1.3.6.1.2.1.10.7.2.1.3	TCP/IP	RCF2665	R/O
dot3StatsSingleCollisionFrames	1.3.6.1.2.1.10.7.2.1.4	TCP/IP	RCF2665	R/O
dot3StatsMultipleCollisionFrames	1.3.6.1.2.1.10.7.2.1.5	TCP/IP	RCF2665	R/O
dot3StatsDeferredTransmissions	1.3.6.1.2.1.10.7.2.1.7	TCP/IP	RCF2665	R/O
dot3StatsLateCollisions	1.3.6.1.2.1.10.7.2.1.8	TCP/IP	RCF2665	R/O
dot3StatsExcessiveCollisions	1.3.6.1.2.1.10.7.2.1.9	TCP/IP	RCF2665	R/O
dot3StatsCarrierSenseErrors	1.3.6.1.2.1.10.7.2.1.11	TCP/IP	RCF2665	R/O
dot3StatsFrameTooLongs	1.3.6.1.2.1.10.7.2.1.13	TCP/IP	RCF2665	R/O
dot3StatsInternalMacReceiveErrors	1.3.6.1.2.1.10.7.2.1.16	TCP/IP	RCF2665	R/O
dot3StatsDuplexStatus	1.3.6.1.2.1.10.7.2.1.19	TCP/IP	RCF2665	R/O
ipoaLisTable	1.3.6.1.2.1.10.46.1.2	TCP/IP	RFC2320	N/A

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipoaLisEntry	1.3.6.1.2.1.10.46.1.2.1	TCP/IP	RFC2320	N/A
ipoaLisSubnetAddr	1.3.6.1.2.1.10.46.1.2.1.1	TCP/IP	RFC2320	R/O
ipoaLisDefaultMtu	1.3.6.1.2.1.10.46.1.2.1.2	TCP/IP	RFC2320	R/O
ipoaLisDefaultEncapsType	1.3.6.1.2.1.10.46.1.2.1.3	TCP/IP	RFC2320	R/O
ipoaLisInactivityTimer	1.3.6.1.2.1.10.46.1.2.1.4	TCP/IP	RFC2320	R/O
ipoaLisMinHoldingTime	1.3.6.1.2.1.10.46.1.2.1.5	TCP/IP	RFC2320	R/O
ipoaLisQDepth	1.3.6.1.2.1.10.46.1.2.1.6	TCP/IP	RFC2320	R/O
ipoaLisMax Calls	1.3.6.1.2.1.10.46.1.2.1.7	TCP/IP	RFC2320	R/O
ipoaLisCacheEntryAge	1.3.6.1.2.1.10.46.1.2.1.8	TCP/IP	RFC2320	R/O
ipoaLisRetries	1.3.6.1.2.1.10.46.1.2.1.9	TCP/IP	RFC2320	R/O
ipoaLisTimeout	1.3.6.1.2.1.10.46.1.2.1.10	TCP/IP	RFC2320	R/O
ipoaLisDefaultPeakCellRate	1.3.6.1.2.1.10.46.1.2.1.11	TCP/IP	RFC2320	R/O
ipoaLisActiveVcs	1.3.6.1.2.1.10.46.1.2.1.12	TCP/IP	RFC2320	R/O
ipoaLisTableInternalMacReceiveErrors	1.3.6.1.2.1.10.46.1.2.1.16	TCP/IP	RFC2320	N/A
ipoaLisIfMappingTable	1.3.6.1.2.1.10.46.1.3	TCP/IP	RFC2320	N/A
ipoaLisIfMappingEntry	1.3.6.1.2.1.10.46.1.3.1	TCP/IP	RFC2320	N/A
ipoaLisIfMappingRowStatus	1.3.6.1.2.1.10.46.1.3.1.1	TCP/IP	RFC2320	R/O
ipoaArpClientTable	1.3.6.1.2.1.10.46.1.4	TCP/IP	RFC2320	N/A
ipoaArpClientEntry	1.3.6.1.2.1.10.46.1.4.1	TCP/IP	RFC2320	N/A
ipoaArpClientAtmAddr	1.3.6.1.2.1.10.46.1.4.1.1	TCP/IP	RFC2320	R/O
ipoaArpClientSrvrInUse	1.3.6.1.2.1.10.46.1.4.1.2	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInReqs	1.3.6.1.2.1.10.46.1.4.1.3	TCP/IP	RFC2320	R/O
ipoaArpClientInArpOutReqs	1.3.6.1.2.1.10.46.1.4.1.4	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInReplies	1.3.6.1.2.1.10.46.1.4.1.5	TCP/IP	RFC2320	R/O
ipoaArpClientInArpOutReplies	1.3.6.1.2.1.10.46.1.4.1.6	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInvalidInReqs	1.3.6.1.2.1.10.46.1.4.1.7	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInvalidOutReqs	1.3.6.1.2.1.10.46.1.4.1.8	TCP/IP	RFC2320	R/O
ipoaArpClientArpInReqs	1.3.6.1.2.1.10.46.1.4.1.9	TCP/IP	RFC2320	R/O
ipoaArpClientArpOutReqs	1.3.6.1.2.1.10.46.1.4.1.10	TCP/IP	RFC2320	R/O
ipoaArpClientArpInReplies	1.3.6.1.2.1.10.46.1.4.1.11	TCP/IP	RFC2320	R/O
ipoaArpClientArpOutReplies	1.3.6.1.2.1.10.46.1.4.1.12	TCP/IP	RFC2320	R/O
ipoaArpClientArpInNaks	1.3.6.1.2.1.10.46.1.4.1.13	TCP/IP	RFC2320	R/O
ipoaArpClientArpOutNaks	1.3.6.1.2.1.10.46.1.4.1.14	TCP/IP	RFC2320	R/O
ipoaArpClientArpUnknownOps	1.3.6.1.2.1.10.46.1.4.1.15	TCP/IP	RFC2320	R/O
ipoaArpClientArpNoSrvrResps	1.3.6.1.2.1.10.46.1.4.1.16	TCP/IP	RFC2320	R/O
ipoaArpRemoteSrvrTable	1.3.6.1.2.1.10.46.1.6	TCP/IP	RFC2320	N/A
ipoaArpRemoteSrvrEntry	1.3.6.1.2.1.10.46.1.6.1	TCP/IP	RFC2320	N/A
ipoaArpRemoteSrvrIpAddr	1.3.6.1.2.1.10.46.1.6.1.4	TCP/IP	RFC2320	R/O
ipoaVcTable	1.3.6.1.2.1.10.46.1.7	TCP/IP	RFC2320	N/A
ipoaVcEntry	1.3.6.1.2.1.10.46.1.7.1	TCP/IP	RFC2320	N/A
ipoaVcType	1.3.6.1.2.1.10.46.1.7.1.3	TCP/IP	RFC2320	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipoaVcNegotiatedEncapsType	1.3.6.1.2.1.10.46.1.7.1.4	TCP/IP	RFC2320	R/O
ipoaVcNegotiatedEncapsMtu	1.3.6.1.2.1.10.46.1.7.1.5	TCP/IP	RFC2320	R/O
ipoaConfigPvcTable	1.3.6.1.2.1.10.46.1.8	TCP/IP	RFC2320	N/A
ipoaConfigPvcEntry	1.3.6.1.2.1.10.46.1.8.1	TCP/IP	RFC2320	N/A
ipoaConfigPvcDefaultMtu	1.3.6.1.2.1.10.46.1.8.1.4	TCP/IP	RFC2320	R/O
snmpInPkts	1.3.6.1.2.1.11.1	Agent	RFC1907	R/O
snmpInBadVersions	1.3.6.1.2.1.11.3	Agent	RFC1907	R/O
snmpInBadCommunityNames	1.3.6.1.2.1.11.4	Agent	RFC1907	R/O
snmpInBadCommunityUses	1.3.6.1.2.1.11.5	Agent	RFC1907	R/O
snmpInASNParseErrs	1.3.6.1.2.1.11.6	Agent	RFC1907	R/O
snmpEnableAuthenTraps	1.3.6.1.2.1.11.30	Agent	RFC1907	R/W
snmpSilentDrops	1.3.6.1.2.1.11.31	Agent	RFC1907	R/O
snmpProxyDrops	1.3.6.1.2.1.11.32	Agent	RFC1907	R/O
ospf	1.3.6.1.2.1.14	omproute	RFC1850	N/A
ospfGeneralGroup	1.3.6.1.2.1.14.1	omproute	RFC1850	N/A
ospfRouterId	1.3.6.1.2.1.14.1.1	omproute	RFC1850	R/O
ospfAdminStat	1.3.6.1.2.1.14.1.2	omproute	RFC1850	R/O
ospfVersionNumber	1.3.6.1.2.1.14.1.3	omproute	RFC1850	R/O
ospfAreaBdrRtrStatus	1.3.6.1.2.1.14.1.4	omproute	RFC1850	R/O
ospfASBdrRtrStatus	1.3.6.1.2.1.14.1.5	omproute	RFC1850	R/O
ospfExternLsaCount	1.3.6.1.2.1.14.1.6	omproute	RFC1850	R/O
ospfExternLsaCksumSum	1.3.6.1.2.1.14.1.7	omproute	RFC1850	R/O
ospfTOSSupport	1.3.6.1.2.1.14.1.8	omproute	RFC1850	R/O
ospfOriginateNewLsas	1.3.6.1.2.1.14.1.9	omproute	RFC1850	R/O
ospfRxNewLsas	1.3.6.1.2.1.14.1.10	omproute	RFC1850	R/O
ospfExtLsdbLimit	1.3.6.1.2.1.14.1.11	omproute	RFC1850	R/O
ospfMulticastExtensions	1.3.6.1.2.1.14.1.12	omproute	RFC1850	R/O
ospfDemandExtensions	1.3.6.1.2.1.14.1.14	omproute	RFC1850	R/O
ospfAreaTable	1.3.6.1.2.1.14.2	omproute	RFC1850	N/A
ospfAreaEntry	1.3.6.1.2.1.14.2.1	omproute	RFC1850	N/A
ospfAreaId	1.3.6.1.2.1.14.2.1.1	omproute	RFC1850	R/O
ospfImportAsExtern	1.3.6.1.2.1.14.2.1.3	omproute	RFC1850	R/O
ospfSpfRuns	1.3.6.1.2.1.14.2.1.4	omproute	RFC1850	R/O
ospfAreaBdrRtrCount	1.3.6.1.2.1.14.2.1.5	omproute	RFC1850	R/O
ospfAsBdrRtrCount	1.3.6.1.2.1.14.2.1.6	omproute	RFC1850	R/O
ospfAreaLsaCount	1.3.6.1.2.1.14.2.1.7	omproute	RFC1850	R/O
ospfAreaLsaCksumSum	1.3.6.1.2.1.14.2.1.8	omproute	RFC1850	R/O
ospfAreaSummary	1.3.6.1.2.1.14.2.1.9	omproute	RFC1850	R/O
ospfStubAreaTable	1.3.6.1.2.1.14.3	omproute	RFC1850	N/A
ospfStubAreaEntry	1.3.6.1.2.1.14.3.1	omproute	RFC1850	N/A
ospfStubAreaId	1.3.6.1.2.1.14.3.1.1	omproute	RFC1850	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ospfStubTOS	1.3.6.1.2.1.14.3.1.2	omproute	RFC1850	R/O
ospfStubMetric	1.3.6.1.2.1.14.3.1.3	omproute	RFC1850	R/O
ospfStubMetricType	1.3.6.1.2.1.14.3.1.5	omproute	RFC1850	R/O
ospfLsdbTable	1.3.6.1.2.1.14.4	omproute	RFC1850	N/A
ospfLsdbEntry	1.3.6.1.2.1.14.4.1	omproute	RFC1850	N/A
ospfLsdbAreaId	1.3.6.1.2.1.14.4.1.1	omproute	RFC1850	R/O
ospfLsdbType	1.3.6.1.2.1.14.4.1.2	omproute	RFC1850	R/O
ospfLsdbLsid	1.3.6.1.2.1.14.4.1.3	omproute	RFC1850	R/O
ospfLsdbRouterId	1.3.6.1.2.1.14.4.1.4	omproute	RFC1850	R/O
ospfLsdbSequence	1.3.6.1.2.1.14.4.1.5	omproute	RFC1850	R/O
ospfLsdbAge	1.3.6.1.2.1.14.4.1.6	omproute	RFC1850	R/O
ospfLsdbChecksum	1.3.6.1.2.1.14.4.1.7	omproute	RFC1850	R/O
ospfLsdbAdvertisement	1.3.6.1.2.1.14.4.1.8	omproute	RFC1850	R/O
ospfIfTable	1.3.6.1.2.1.14.7	omproute	RFC1850	N/A
ospfIfEntry	1.3.6.1.2.1.14.7.1	omproute	RFC1850	N/A
ospfIfIpAddress	1.3.6.1.2.1.14.7.1.1	omproute	RFC1850	R/O
ospfAddressLessIf	1.3.6.1.2.1.14.7.1.2	omproute	RFC1850	R/O
ospfIfAreaId	1.3.6.1.2.1.14.7.1.3	omproute	RFC1850	R/O
ospfIfType	1.3.6.1.2.1.14.7.1.4	omproute	RFC1850	R/O
ospfIfAdminStat	1.3.6.1.2.1.14.7.1.5	omproute	RFC1850	R/O
ospfIfRtrPriority	1.3.6.1.2.1.14.7.1.6	omproute	RFC1850	R/O
ospfIfTransitDelay	1.3.6.1.2.1.14.7.1.7	omproute	RFC1850	R/O
ospfIfRetransInterval	1.3.6.1.2.1.14.7.1.8	omproute	RFC1850	R/O
ospfIfHelloInterval	1.3.6.1.2.1.14.7.1.9	omproute	RFC1850	R/O
ospfIfRtrDeadInterval	1.3.6.1.2.1.14.7.1.10	omproute	RFC1850	R/O
ospfIfPollInterval	1.3.6.1.2.1.14.7.1.11	omproute	RFC1850	R/O
ospfIfState	1.3.6.1.2.1.14.7.1.12	omproute	RFC1850	R/O
ospfIfDesignatedRouter	1.3.6.1.2.1.14.7.1.13	omproute	RFC1850	R/O
ospfIfBackupDesignatedRouter	1.3.6.1.2.1.14.7.1.14	omproute	RFC1850	R/O
ospfIfEvents	1.3.6.1.2.1.14.7.1.15	omproute	RFC1850	R/O
ospfIfAuthKey	1.3.6.1.2.1.14.7.1.16	omproute	RFC1850	R/O
ospfIfMulticastForwarding	1.3.6.1.2.1.14.7.1.18	omproute	RFC1850	R/O
ospfIfDemand	1.3.6.1.2.1.14.7.1.19	omproute	RFC1850	R/O
ospfIfAuthType	1.3.6.1.2.1.14.7.1.20	omproute	RFC1850	R/O
ospfIfMetricTable	1.3.6.1.2.1.14.8	omproute	RFC1850	N/A
ospfIfMetricEntry	1.3.6.1.2.1.14.8.1	omproute	RFC1850	N/A
ospfIfMetricIpAddress	1.3.6.1.2.1.14.8.1.1	omproute	RFC1850	R/O
ospfIfMetricAddressLessIf	1.3.6.1.2.1.14.8.1.2	omproute	RFC1850	R/O
ospfIfMetricTOS	1.3.6.1.2.1.14.8.1.3	omproute	RFC1850	R/O
ospfIfMetricValue	1.3.6.1.2.1.14.8.1.4	omproute	RFC1850	R/O
ospfVirtIfTable	1.3.6.1.2.1.14.9	omproute	RFC1850	N/A

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ospfVirtIfEntry	1.3.6.1.2.1.14.9.1	omproute	RFC1850	N/A
ospfVirtIfAreaId	1.3.6.1.2.1.14.9.1.1	omproute	RFC1850	R/O
ospfVirtIfNeighbor	1.3.6.1.2.1.14.9.1.2	omproute	RFC1850	R/O
ospfVirtIfTransitDelay	1.3.6.1.2.1.14.9.1.3	omproute	RFC1850	R/O
ospfVirtIfRetransInterval	1.3.6.1.2.1.14.9.1.4	omproute	RFC1850	R/O
ospfVirtIfHelloInterval	1.3.6.1.2.1.14.9.1.5	omproute	RFC1850	R/O
ospfVirtIfRtrDeadInterval	1.3.6.1.2.1.14.9.1.6	omproute	RFC1850	R/O
ospfVirtIfState	1.3.6.1.2.1.14.9.1.7	omproute	RFC1850	R/O
ospfVirtIfEvents	1.3.6.1.2.1.14.9.1.8	omproute	RFC1850	R/O
ospfVirtIfAuthKey	1.3.6.1.2.1.14.9.1.9	omproute	RFC1850	R/O
ospfVirtIfAuthType	1.3.6.1.2.1.14.9.1.11	omproute	RFC1850	R/O
ospfNbrTable	1.3.6.1.2.1.14.10	omproute	RFC1850	N/A
ospfNbrEntry	1.3.6.1.2.1.14.10.1	omproute	RFC1850	N/A
ospfNbrIpAddr	1.3.6.1.2.1.14.10.1.1	omproute	RFC1850	R/O
ospfNbrAddressLessIndex	1.3.6.1.2.1.14.10.1.2	omproute	RFC1850	R/O
ospfNbrRtrId	1.3.6.1.2.1.14.10.1.3	omproute	RFC1850	R/O
ospfNbrOptions	1.3.6.1.2.1.14.10.1.4	omproute	RFC1850	R/O
ospfNbrPriority	1.3.6.1.2.1.14.10.1.5	omproute	RFC1850	R/O
ospfNbrState	1.3.6.1.2.1.14.10.1.6	omproute	RFC1850	R/O
ospfNbrEvents	1.3.6.1.2.1.14.10.1.7	omproute	RFC1850	R/O
ospfNbrLsRetransQLen	1.3.6.1.2.1.14.10.1.8	omproute	RFC1850	R/O
ospfNbmaNbrPermanence	1.3.6.1.2.1.14.10.1.10	omproute	RFC1850	R/O
ospfNbrHelloSuppressed	1.3.6.1.2.1.14.10.1.11	omproute	RFC1850	R/O
ospfVirtNbrTable	1.3.6.1.2.1.14.11	omproute	RFC1850	N/A
ospfVirtNbrEntry	1.3.6.1.2.1.14.11.1	omproute	RFC1850	N/A
ospfVirtNbrArea	1.3.6.1.2.1.14.11.1.1	omproute	RFC1850	R/O
ospfVirtNbrRtrId	1.3.6.1.2.1.14.11.1.2	omproute	RFC1850	R/O
ospfVirtNbrIpAddr	1.3.6.1.2.1.14.11.1.3	omproute	RFC1850	R/O
ospfVirtNbrOptions	1.3.6.1.2.1.14.11.1.4	omproute	RFC1850	R/O
ospfVirtNbrState	1.3.6.1.2.1.14.11.1.5	omproute	RFC1850	R/O
ospfVirtNbrEvents	1.3.6.1.2.1.14.11.1.6	omproute	RFC1850	R/O
ospfVirtNbrLsRetransQLen	1.3.6.1.2.1.14.11.1.7	omproute	RFC1850	R/O
ospfVirtNbrHelloSuppressed	1.3.6.1.2.1.14.11.1.8	omproute	RFC1850	R/O
ospfExtLsdbTable	1.3.6.1.2.1.14.12	omproute	RFC1850	N/A
ospfExtLsdbEntry	1.3.6.1.2.1.14.12.1	omproute	RFC1850	N/A
ospfExtLsdbType	1.3.6.1.2.1.14.12.1.1	omproute	RFC1850	R/O
ospfExtLsdbLsid	1.3.6.1.2.1.14.12.1.2	omproute	RFC1850	R/O
ospfExtLsdbRouterId	1.3.6.1.2.1.14.12.1.3	omproute	RFC1850	R/O
ospfExtLsdbSequence	1.3.6.1.2.1.14.12.1.4	omproute	RFC1850	R/O
ospfExtLsdbAge	1.3.6.1.2.1.14.12.1.5	omproute	RFC1850	R/O
ospfExtLsdbChecksum	1.3.6.1.2.1.14.12.1.6	omproute	RFC1850	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ospfExtLsdbAdvertisement	1.3.6.1.2.1.14.12.1.7	omproute	RFC1850	R/O
ospfAreaAggregateTable	1.3.6.1.2.1.14.14	omproute	RFC1850	N/A
ospfAreaAggregateEntry	1.3.6.1.2.1.14.14.1	omproute	RFC1850	N/A
ospfAreaAggregateAreaID	1.3.6.1.2.1.14.14.1.1	omproute	RFC1850	R/O
ospfAreaAggregateLsdbType	1.3.6.1.2.1.14.14.1.2	omproute	RFC1850	R/O
ospfAreaAggregateNet	1.3.6.1.2.1.14.14.1.3	omproute	RFC1850	R/O
ospfAreaAggregateMask	1.3.6.1.2.1.14.14.1.4	omproute	RFC1850	R/O
ospfAreaAggregateEffect	1.3.6.1.2.1.14.14.1.6	omproute	RFC1850	R/O
ifXTable	1.3.6.1.2.1.31.1.1	TCP/IP	RFC2233	N/A
ifXEntry	1.3.6.1.2.1.31.1.1.1	TCP/IP	RFC2233	N/A
ifName	1.3.6.1.2.1.31.1.1.1.1	TCP/IP	RFC2233	R/O
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	TCP/IP	RFC2233	R/O
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3	TCP/IP	RFC2233	R/O
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	TCP/IP	RFC2233	R/O
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5	TCP/IP	RFC2233	R/O
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	TCP/IP	RFC2233	R/O
ifHCInUcastPkts	1.3.6.1.2.1.31.1.1.1.7	TCP/IP	RFC2233	R/O
ifHCInMulticastPkts	1.3.6.1.2.1.31.1.1.1.8	TCP/IP	RFC2233	R/O
ifHCInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.9	TCP/IP	RFC2233	R/O
ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	TCP/IP	RFC2233	R/O
ifHCOutUcastPkts	1.3.6.1.2.1.31.1.1.1.11	TCP/IP	RFC2233	R/O
ifHCOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.12	TCP/IP	RFC2233	R/O
ifHCOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.13	TCP/IP	RFC2233	R/O
ifLinkUpDownTrapEnable	1.3.6.1.2.1.31.1.1.1.14	TCP/IP	RFC2233	R/W
ifHighSpeed	1.3.6.1.2.1.31.1.1.1.15	TCP/IP	RFC2233	R/O
ifPromiscuousMode	1.3.6.1.2.1.31.1.1.1.16	TCP/IP	RFC2233	R/O
ifConnectorPresent	1.3.6.1.2.1.31.1.1.1.17	TCP/IP	RFC2233	R/O
ifAlias	1.3.6.1.2.1.31.1.1.1.18	TCP/IP	RFC2233	R/W
ifCounterDiscontinuityTime	1.3.6.1.2.1.31.1.1.1.19	TCP/IP	RFC2233	R/O
ifStackTable	1.3.6.1.2.1.31.1.2	TCP/IP	RFC2233	N/A
ifStackEntry	1.3.6.1.2.1.31.1.2.1	TCP/IP	RFC2233	N/A
ifStackStatus	1.3.6.1.2.1.31.1.2.1.3	TCP/IP	RFC2233	R/O
atmInterfaceConfTable	1.3.6.1.2.1.37.1.2	TCP/IP	RFC1695	N/A
atmInterfaceConfEntry	1.3.6.1.2.1.37.1.2.1	TCP/IP	RFC1695	N/A
atmInterfaceMaxVpcs	1.3.6.1.2.1.37.1.2.1.1	TCP/IP	RFC1695	R/O
atmInterfaceMaxVccs	1.3.6.1.2.1.37.1.2.1.2	TCP/IP	RFC1695	R/O
atmInterfaceConfVpcs	1.3.6.1.2.1.37.1.2.1.3	TCP/IP	RFC1695	R/O
atmInterfaceConfVccs	1.3.6.1.2.1.37.1.2.1.4	TCP/IP	RFC1695	R/O
atmInterfaceMaxActiveVpiBits	1.3.6.1.2.1.37.1.2.1.5	TCP/IP	RFC1695	R/O
atmInterfaceMaxActiveVciBits	1.3.6.1.2.1.37.1.2.1.6	TCP/IP	RFC1695	R/O
atmInterfaceIlmiVpi	1.3.6.1.2.1.37.1.2.1.7	TCP/IP	RFC1695	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
atmInterfaceIlmiVci	1.3.6.1.2.1.37.1.2.1.8	TCP/IP	RFC1695	R/O
atmInterfaceAddressType	1.3.6.1.2.1.37.1.2.1.9	TCP/IP	RFC1695	R/O
atmInterfaceAdminAddress	1.3.6.1.2.1.37.1.2.1.10	TCP/IP	RFC1695	R/O
atmInterfaceMyNeighborIpAddress	1.3.6.1.2.1.37.1.2.1.11	TCP/IP	RFC1695	R/O
atmInterfaceMyNeighborIfName	1.3.6.1.2.1.37.1.2.1.12	TCP/IP	RFC1695	R/O
dpiPort	1.3.6.1.4.1.2.2.1.1.0	Agent	RFC1592	R/O
dpiPortForTCP	1.3.6.1.4.1.2.2.1.1.1.0	Agent	RFC1592	R/O
dpiPortForUDP	1.3.6.1.4.1.2.2.1.1.2.0	Agent	RFC1592	R/O
dpiPathNameForUnixStream	1.3.6.1.4.1.2.2.1.1.3.0	Agent	RFC1592B	R/O
saDefaultTimeout	1.3.6.1.4.1.2.4.12.1	Agent	SAMIB	R/W
saMaxTimeout	1.3.6.1.4.1.2.4.12.2	Agent	SAMIB	R/W
saAllowDuplicateIDs	1.3.6.1.4.1.2.4.12.3	Agent	SAMIB	R/W
saNumber	1.3.6.1.4.1.2.4.12.4	Agent	SAMIB	R/O
saAllPacketsIn	1.3.6.1.4.1.2.4.12.5	Agent	SAMIB	R/O
saAllPacketsOut	1.3.6.1.4.1.2.4.12.6	Agent	SAMIB	R/O
saTable	1.3.6.1.4.1.2.4.12.7	Agent	SAMIB	N/A
saEntry	1.3.6.1.4.1.2.4.12.7.1	Agent	SAMIB	N/A
saIndex	1.3.6.1.4.1.2.4.12.7.1.1	Agent	SAMIB	R/O
saIdentifier	1.3.6.1.4.1.2.4.12.7.1.2	Agent	SAMIB	R/O
saDescription	1.3.6.1.4.1.2.4.12.7.1.3	Agent	SAMIB	R/O
saStatus	1.3.6.1.4.1.2.4.12.7.1.4	Agent	SAMIB	R/W
saStatusChangeTime	1.3.6.1.4.1.2.4.12.7.1.5	Agent	SAMIB	R/O
saProtocol	1.3.6.1.4.1.2.4.12.7.1.6	Agent	SAMIB	R/O
saProtocolVersion	1.3.6.1.4.1.2.4.12.7.1.7	Agent	SAMIB	R/O
saProtocolRelease	1.3.6.1.4.1.2.4.12.7.1.8	Agent	SAMIB	R/O
saTransport	1.3.6.1.4.1.2.4.12.7.1.9	Agent	SAMIB	R/O
saTransportAddress	1.3.6.1.4.1.2.4.12.7.1.10	Agent	SAMIB	R/O
saTimeout	1.3.6.1.4.1.2.4.12.7.1.11	Agent	SAMIB	R/W
saMaxVarBinds	1.3.6.1.4.1.2.4.12.7.1.12	Agent	SAMIB	R/O
saPacketsIn	1.3.6.1.4.1.2.4.12.7.1.13	Agent	SAMIB	R/O
saPacketsOut	1.3.6.1.4.1.2.4.12.7.1.14	Agent	SAMIB	R/O
saTreeTable	1.3.6.1.4.1.2.4.12.8	Agent	SAMIB	N/A
saTreeEntry	1.3.6.1.4.1.2.4.12.8.1	Agent	SAMIB	N/A
saTsubtree	1.3.6.1.4.1.2.4.12.8.1.1	Agent	SAMIB	R/O
saTpriority	1.3.6.1.4.1.2.4.12.8.1.2	Agent	SAMIB	R/O
saTindex	1.3.6.1.4.1.2.4.12.8.1.3	Agent	SAMIB	R/O
saTstatus	1.3.6.1.4.1.2.4.12.8.1.4	Agent	SAMIB	R/W
saTtimeout	1.3.6.1.4.1.2.4.12.8.1.5	Agent	SAMIB	R/W
slapm2PolicyUpdates	1.3.6.1.4.1.2.5.30.1.1.1	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PolicyLastUpdated	1.3.6.1.4.1.2.5.30.1.1.2	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PolicyLastChecked	1.3.6.1.4.1.2.5.30.1.1.3	NSLAPM2	NETWORK-SLAPM2-MIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
slapm2PolicyDeletedTrapEnable	1.3.6.1.4.1.2.5.30.1.1.4	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PolicyMonInterval	1.3.6.1.4.1.2.5.30.1.1.5	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PolicyRuleTable	1.3.6.1.4.1.2.5.30.1.2.1	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PolicyRuleEntry	1.3.6.1.4.1.2.5.30.1.2.1.1	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PolicyRuleIndex	1.3.6.1.4.1.2.5.30.1.2.1.1.1	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PolicyRuleNameOfRule	1.3.6.1.4.1.2.5.30.1.2.1.1.2	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PolicyRuleOperStatus	1.3.6.1.4.1.2.5.30.1.2.1.1.3	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PolicyRuleDeleteTime	1.3.6.1.4.1.2.5.30.1.2.1.1.4	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PolicyRuleStatsTable	1.3.6.1.4.1.2.5.30.1.2.2	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PolicyRuleStatsEntry	1.3.6.1.4.1.2.5.30.1.2.2.1	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PRStatsActiveConns	1.3.6.1.4.1.2.5.30.1.2.2.1.1	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsAcceptedConns	1.3.6.1.4.1.2.5.30.1.2.2.1.2	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsDeniedConns	1.3.6.1.4.1.2.5.30.1.2.2.1.3	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsLActivated	1.3.6.1.4.1.2.5.30.1.2.2.1.4	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsLastMapping	1.3.6.1.4.1.2.5.30.1.2.2.1.5	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsInOctets	1.3.6.1.4.1.2.5.30.1.2.2.1.6	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsInInProOctets	1.3.6.1.4.1.2.5.30.1.2.2.1.7	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsOutOctets	1.3.6.1.4.1.2.5.30.1.2.2.1.8	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsOutInProOctets	1.3.6.1.4.1.2.5.30.1.2.2.1.9	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsInPackets	1.3.6.1.4.1.2.5.30.1.2.2.1.10	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsInInProPackets	1.3.6.1.4.1.2.5.30.1.2.2.1.11	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsOutPackets	1.3.6.1.4.1.2.5.30.1.2.2.1.12	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsOutInProPackets	1.3.6.1.4.1.2.5.30.1.2.2.1.13	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsAvgTcpRtt	1.3.6.1.4.1.2.5.30.1.2.2.1.14	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsMDTcpRtt	1.3.6.1.4.1.2.5.30.1.2.2.1.15	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsAvgAcceptQDelay	1.3.6.1.4.1.2.5.30.1.2.2.1.16	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsMDAcceptQDelay	1.3.6.1.4.1.2.5.30.1.2.2.1.17	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsAvgSrvrReactTime	1.3.6.1.4.1.2.5.30.1.2.2.1.18	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsMDSrvrReactTime	1.3.6.1.4.1.2.5.30.1.2.2.1.19	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsTcpReXmitOctets	1.3.6.1.4.1.2.5.30.1.2.2.1.20	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsTcpReXmitPackets	1.3.6.1.4.1.2.5.30.1.2.2.1.21	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRStatsTcpReXmitTimeouts	1.3.6.1.4.1.2.5.30.1.2.2.1.22	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRMonTable	1.3.6.1.4.1.2.5.30.1.2.6	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PRMonEntry	1.3.6.1.4.1.2.5.30.1.2.6.1	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PRMonOwnerIndex	1.3.6.1.4.1.2.5.30.1.2.6.1.1	NSLAPM2	NETWORK-SLAPM2-MIB	N/A
slapm2PRMonTrapEnable	1.3.6.1.4.1.2.5.30.1.2.6.1.2	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonStatus	1.3.6.1.4.1.2.5.30.1.2.6.1.3	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRMonTrapFilter	1.3.6.1.4.1.2.5.30.1.2.6.1.4	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonIntTime	1.3.6.1.4.1.2.5.30.1.2.6.1.5	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRMonTcpRttDelayHigh	1.3.6.1.4.1.2.5.30.1.2.6.1.6	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonTcpRttDelayLow	1.3.6.1.4.1.2.5.30.1.2.6.1.7	NSLAPM2	NETWORK-SLAPM2-MIB	R/C

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
slapm2PRMonTcpRttCurrentDelay	1.3.6.1.4.1.2.5.30.1.2.6.1.8	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRMonTcpReXmitHigh	1.3.6.1.4.1.2.5.30.1.2.6.1.9	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonTcpReXmitLow	1.3.6.1.4.1.2.5.30.1.2.6.1.10	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonCurrentTcpReXmit	1.3.6.1.4.1.2.5.30.1.2.6.1.11	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRMonAcceptQDelayHigh	1.3.6.1.4.1.2.5.30.1.2.6.1.12	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonAcceptQDelayLow	1.3.6.1.4.1.2.5.30.1.2.6.1.13	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
slapm2PRMonAcceptQCurrentDelay	1.3.6.1.4.1.2.5.30.1.2.6.1.14	NSLAPM2	NETWORK-SLAPM2-MIB	R/O
slapm2PRMonRowStatus	1.3.6.1.4.1.2.5.30.1.2.6.1.15	NSLAPM2	NETWORK-SLAPM2-MIB	R/C
ibm3172Descr	1.3.6.1.4.1.2.6.1.1.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172Contact	1.3.6.1.4.1.2.6.1.1.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172Location	1.3.6.1.4.1.2.6.1.1.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifNumber	1.3.6.1.4.1.2.6.1.1.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifTrapEnable	1.3.6.1.4.1.2.6.1.2.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifInChanOctets	1.3.6.1.4.1.2.6.1.3.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutChanOctets	1.3.6.1.4.1.2.6.1.3.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifInChanBlocks	1.3.6.1.4.1.2.6.1.3.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutChanBlocks	1.3.6.1.4.1.2.6.1.3.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANOctets	1.3.6.1.4.1.2.6.1.4.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANOctets	1.3.6.1.4.1.2.6.1.4.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANFrames	1.3.6.1.4.1.2.6.1.4.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANFrames	1.3.6.1.4.1.2.6.1.4.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANErrors	1.3.6.1.4.1.2.6.1.4.1.5	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANErrors	1.3.6.1.4.1.2.6.1.4.1.6	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANDiscards	1.3.6.1.4.1.2.6.1.4.1.7	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANDiscards	1.3.6.1.4.1.2.6.1.4.1.8	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkRcvOctets	1.3.6.1.4.1.2.6.1.5.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkXmitOctets	1.3.6.1.4.1.2.6.1.5.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkRcvFrames	1.3.6.1.4.1.2.6.1.5.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkXmitBlocks	1.3.6.1.4.1.2.6.1.5.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifInBlkErrors	1.3.6.1.4.1.2.6.1.5.1.5	TCP/IP	ibm3172MIB	R/O
ibm3172ifInBlkDiscards	1.3.6.1.4.1.2.6.1.5.1.6	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkRcvOctets	1.3.6.1.4.1.2.6.1.6.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkXmitOctets	1.3.6.1.4.1.2.6.1.6.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkRcvBlocks	1.3.6.1.4.1.2.6.1.6.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkXmitFrames	1.3.6.1.4.1.2.6.1.6.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutDblkErrors	1.3.6.1.4.1.2.6.1.6.1.5	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutDblkDiscards	1.3.6.1.4.1.2.6.1.6.1.6	TCP/IP	ibm3172MIB	R/O
ibm3172ifDeviceNumber	1.3.6.1.4.1.2.6.1.7.1.1	TCP/IP	ibm3172MIB	R/O
ibmRemotePingTable	1.3.6.1.4.1.2.6.19.2.2.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmRemotePingEntry	1.3.6.1.4.1.2.6.19.2.2.1.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRPingResponseTime	1.3.6.1.4.1.2.6.19.2.2.1.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmTcpipMvsRemPingTable	1.3.6.1.4.1.2.6.19.2.2.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsRemPingEntry	1.3.6.1.4.1.2.6.19.2.2.1.2.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRemPingPacketSize	1.3.6.1.4.1.2.6.19.2.2.1.2.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRemPingTimeOut	1.3.6.1.4.1.2.6.19.2.2.1.2.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRemPingHostAddrType	1.3.6.1.4.1.2.6.19.2.2.1.2.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRemPingHostAddr	1.3.6.1.4.1.2.6.19.2.2.1.2.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRemPingResponseTime	1.3.6.1.4.1.2.6.19.2.2.1.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSubagentCacheTime	1.3.6.1.4.1.2.6.19.2.2.2.1	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIgnoreRedirect	1.3.6.1.4.1.2.6.19.2.2.2.2	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsArpCacheTimeout	1.3.6.1.4.1.2.6.19.2.2.2.3	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpKeepAliveTimer	1.3.6.1.4.1.2.6.19.2.2.2.4	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpReceiveBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.5	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpSendBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.6	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpChecksum	1.3.6.1.4.1.2.6.19.2.2.2.7	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIplDateAndTime	1.3.6.1.4.1.2.6.19.2.2.2.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsNoUdpQueueLimit	1.3.6.1.4.1.2.6.19.2.2.2.9	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsSoMaxConn	1.3.6.1.4.1.2.6.19.2.2.2.10	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpipProcname	1.3.6.1.4.1.2.6.19.2.2.2.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpipAsid	1.3.6.1.4.1.2.6.19.2.2.2.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSourceVipaEnabled	1.3.6.1.4.1.2.6.19.2.2.2.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsasfSysplexName	1.3.6.1.4.1.2.6.19.2.2.2.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsasfHostName	1.3.6.1.4.1.2.6.19.2.2.2.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsasfProductVersion	1.3.6.1.4.1.2.6.19.2.2.2.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPrimaryInterfaceIfIndex	1.3.6.1.4.1.2.6.19.2.2.2.17	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIpMaxReassemblySize	1.3.6.1.4.1.2.6.19.2.2.2.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpRestrictLowPorts	1.3.6.1.4.1.2.6.19.2.2.2.19	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpRestrictLowPorts	1.3.6.1.4.1.2.6.19.2.2.2.20	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpSendBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.21	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpRecvBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.22	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpipStatisticsEnabled	1.3.6.1.4.1.2.6.19.2.2.2.23	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsMaximumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.2.25	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsMinimumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.2.26	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRoundTripGain	1.3.6.1.4.1.2.6.19.2.2.2.27	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsVarianceGain	1.3.6.1.4.1.2.6.19.2.2.2.28	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsVarianceMultiplier	1.3.6.1.4.1.2.6.19.2.2.2.29	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSendGarbageEnabled	1.3.6.1.4.1.2.6.19.2.2.2.30	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpMaxReceiveBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.31	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsPathMtuDscEnabled	1.3.6.1.4.1.2.6.19.2.2.2.33	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsMultipathType	1.3.6.1.4.1.2.6.19.2.2.2.34	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIpForwarding	1.3.6.1.4.1.2.6.19.2.2.2.35	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsDevRetryDuration	1.3.6.1.4.1.2.6.19.2.2.2.36	TCP/IP	ibmTCPIPmvsMIB	R/W

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsTcpFinwait2Time	1.3.6.1.4.1.2.6.19.2.2.2.37	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpTimeStamp	1.3.6.1.4.1.2.6.19.2.2.2.38	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpipSubagentVersion	1.3.6.1.4.1.2.6.19.2.2.2.39	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSystemName	1.3.6.1.4.1.2.6.19.2.2.2.40	TCP/IP	ibmTCPIPmvsMIB	R/O
bmMvsSysplexName	1.3.6.1.4.1.2.6.19.2.2.2.41	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIp6Forwarding	1.3.6.1.4.1.2.6.19.2.2.2.42	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIp6IcmpErrorLimit	1.3.6.1.4.1.2.6.19.2.2.2.43	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIp6IgnoreRedirect	1.3.6.1.4.1.2.6.19.2.2.2.44	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIp6IgnoreRtrHopLimit	1.3.6.1.4.1.2.6.19.2.2.2.45	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIp6MultipathType	1.3.6.1.4.1.2.6.19.2.2.2.46	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIp6SourceVipaEnabled	1.3.6.1.4.1.2.6.19.2.2.2.47	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIp6TcpStackSrcVipaIntfName	1.3.6.1.4.1.2.6.19.2.2.2.48	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpsecEnabled	1.3.6.1.4.1.2.6.19.2.2.2.49	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpTtlsEnabled	1.3.6.1.4.1.2.6.19.2.2.2.50	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpipXcfGroupName	1.3.6.1.4.1.2.6.19.2.2.2.51	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIp6IpsecEnabled	1.3.6.1.4.1.2.6.19.2.2.2.52	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsCpcNd	1.3.6.1.4.1.2.6.19.2.2.2.53	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpSelectiveAck	1.3.6.1.4.1.2.6.19.2.2.2.54	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpEphemeralPortLow	1.3.6.1.4.1.2.6.19.2.2.2.55	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpEphemeralPortHigh	1.3.6.1.4.1.2.6.19.2.2.2.56	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEphemeralPortLow	1.3.6.1.4.1.2.6.19.2.2.2.57	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEphemeralPortHigh	1.3.6.1.4.1.2.6.19.2.2.2.58	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectInitInterval	1.3.6.1.4.1.2.6.19.2.2.2.59	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectTimeout	1.3.6.1.4.1.2.6.19.2.2.2.60	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpFrrThreshold	1.3.6.1.4.1.2.6.19.2.2.2.61	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpKeepAliveProbeInterval	1.3.6.1.4.1.2.6.19.2.2.2.62	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpKeepAliveProbesNum	1.3.6.1.4.1.2.6.19.2.2.2.63	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpNagle	1.3.6.1.4.1.2.6.19.2.2.2.64	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpQueuedRtt	1.3.6.1.4.1.2.6.19.2.2.2.65	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpRetransmitAttempts	1.3.6.1.4.1.2.6.19.2.2.2.66	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpMaxSendBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.67	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpTimewaitInterval	1.3.6.1.4.1.2.6.19.2.2.2.68	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceType	1.3.6.1.4.1.2.6.19.2.2.3.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceBaseNumber	1.3.6.1.4.1.2.6.19.2.2.3.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceIoBufferSize	1.3.6.1.4.1.2.6.19.2.2.3.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceAutoRestart	1.3.6.1.4.1.2.6.19.2.2.3.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceNetmanEnabled	1.3.6.1.4.1.2.6.19.2.2.3.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceHostClawName	1.3.6.1.4.1.2.6.19.2.2.3.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceWorkstationClawName	1.3.6.1.4.1.2.6.19.2.2.3.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceReadBuffers	1.3.6.1.4.1.2.6.19.2.2.3.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceReadSize	1.3.6.1.4.1.2.6.19.2.2.3.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsDeviceWriteBuffers	1.3.6.1.4.1.2.6.19.2.2.3.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceWriteSize	1.3.6.1.4.1.2.6.19.2.2.3.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceProcname	1.3.6.1.4.1.2.6.19.2.2.3.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceIncomingSvcEnabled	1.3.6.1.4.1.2.6.19.2.2.3.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceLuName	1.3.6.1.4.1.2.6.19.2.2.3.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceRouterStatus	1.3.6.1.4.1.2.6.19.2.2.3.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceActualRouterStatus	1.3.6.1.4.1.2.6.19.2.2.3.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceConfigPackingMode	1.3.6.1.4.1.2.6.19.2.2.3.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceActualPackingMode	1.3.6.1.4.1.2.6.19.2.2.3.1.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkType	1.3.6.1.4.1.2.6.19.2.2.3.2.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkDeviceIndex	1.3.6.1.4.1.2.6.19.2.2.3.2.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkAdapterAddr	1.3.6.1.4.1.2.6.19.2.2.3.2.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkNumber	1.3.6.1.4.1.2.6.19.2.2.3.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkIbmtrCanonical	1.3.6.1.4.1.2.6.19.2.2.3.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkIbmtrBcast	1.3.6.1.4.1.2.6.19.2.2.3.2.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkMcast	1.3.6.1.4.1.2.6.19.2.2.3.2.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkChecksumEnabled	1.3.6.1.4.1.2.6.19.2.2.3.2.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkArpSupport	1.3.6.1.4.1.2.6.19.2.2.3.2.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkMacAddress	1.3.6.1.4.1.2.6.19.2.2.3.2.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkVlanId	1.3.6.1.4.1.2.6.19.2.2.3.2.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkVlanPriorityEnabled	1.3.6.1.4.1.2.6.19.2.2.3.2.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkReadStorageSize	1.3.6.1.4.1.2.6.19.2.2.3.2.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkInboundPerfType	1.3.6.1.4.1.2.6.19.2.2.3.2.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkChecksumOffloadEnabled	1.3.6.1.4.1.2.6.19.2.2.3.2.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkMcastRefCount	1.3.6.1.4.1.2.6.19.2.2.3.3.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTCPIPMvsPktTraceTable	1.3.6.1.4.1.2.6.19.2.2.3.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTCPIPMvsPktTraceEntry	1.3.6.1.4.1.2.6.19.2.2.3.4.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceProto	1.3.6.1.4.1.2.6.19.2.2.3.4.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceSrcPort	1.3.6.1.4.1.2.6.19.2.2.3.4.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceDestPort	1.3.6.1.4.1.2.6.19.2.2.3.4.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceIpAddrType	1.3.6.1.4.1.2.6.19.2.2.3.4.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceIpAddr	1.3.6.1.4.1.2.6.19.2.2.3.4.1.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceIpAddrPrefixLen	1.3.6.1.4.1.2.6.19.2.2.3.4.1.6	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsPktTraceLen	1.3.6.1.4.1.2.6.19.2.2.3.4.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPktTraceIntfName	1.3.6.1.4.1.2.6.19.2.2.3.4.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPktTraceRecCount	1.3.6.1.4.1.2.6.19.2.2.3.4.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsIfTable	1.3.6.1.4.1.2.6.19.2.2.3.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsIfEntry	1.3.6.1.4.1.2.6.19.2.2.3.5.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsIfType	1.3.6.1.4.1.2.6.19.2.2.3.5.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfDeviceIndex	1.3.6.1.4.1.2.6.19.2.2.3.5.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfFlag	1.3.6.1.4.1.2.6.19.2.2.3.5.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsIfNumber	1.3.6.1.4.1.2.6.19.2.2.3.5.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfIbmtrBcast	1.3.6.1.4.1.2.6.19.2.2.3.5.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfArpSupport	1.3.6.1.4.1.2.6.19.2.2.3.5.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfConfigRouterStatus	1.3.6.1.4.1.2.6.19.2.2.3.5.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfActualRouterStatus	1.3.6.1.4.1.2.6.19.2.2.3.5.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfDupAddrDetCount	1.3.6.1.4.1.2.6.19.2.2.3.5.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfSrcVipaIntfName	1.3.6.1.4.1.2.6.19.2.2.3.5.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfConfigMtu	1.3.6.1.4.1.2.6.19.2.2.3.5.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfVlanId	1.3.6.1.4.1.2.6.19.2.2.3.5.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfReadStorageSize	1.3.6.1.4.1.2.6.19.2.2.3.5.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfInboundPerfType	1.3.6.1.4.1.2.6.19.2.2.3.5.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfChpid	1.3.6.1.4.1.2.6.19.2.2.3.5.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfSecClass	1.3.6.1.4.1.2.6.19.2.2.3.5.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfMonSysplexStatus	1.3.6.1.4.1.2.6.19.2.2.3.5.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfDatapath	1.3.6.1.4.1.2.6.19.2.2.3.5.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfTrleName	1.3.6.1.4.1.2.6.19.2.2.3.5.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIfPNetID	1.3.6.1.4.1.2.6.19.2.2.3.5.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsIfMcastTable	1.3.6.1.4.1.2.6.19.2.2.3.6	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsIfMcastEntry	1.3.6.1.4.1.2.6.19.2.2.3.6.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsIfMcastAddrType	1.3.6.1.4.1.2.6.19.2.2.3.6.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsIfMcastAddr	1.3.6.1.4.1.2.6.19.2.2.3.6.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsIfMcastRefCount	1.3.6.1.4.1.2.6.19.2.2.3.6.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortNumberLow	1.3.6.1.4.1.2.6.19.2.2.4.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortNumberHigh	1.3.6.1.4.1.2.6.19.2.2.4.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortProtocol	1.3.6.1.4.1.2.6.19.2.2.4.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortProcName	1.3.6.1.4.1.2.6.19.2.2.4.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortAutoLoggable	1.3.6.1.4.1.2.6.19.2.2.4.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortDelayAcks	1.3.6.1.4.1.2.6.19.2.2.4.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortOptMaxSegmentSize	1.3.6.1.4.1.2.6.19.2.2.4.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortSharePort	1.3.6.1.4.1.2.6.19.2.2.4.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortBindIpAddr	1.3.6.1.4.1.2.6.19.2.2.4.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortSAFResource	1.3.6.1.4.1.2.6.19.2.2.4.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortReuse	1.3.6.1.4.1.2.6.19.2.2.4.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortBindIpAddressType	1.3.6.1.4.1.2.6.19.2.2.4.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortBindIpAddress	1.3.6.1.4.1.2.6.19.2.2.4.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortSharePortWlm	1.3.6.1.4.1.2.6.19.2.2.4.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortNoSmcr	1.3.6.1.4.1.2.6.19.2.2.4.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayMaximumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.5.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayMinimumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.5.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayRoundTripGain	1.3.6.1.4.1.2.6.19.2.2.5.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayVarianceGain	1.3.6.1.4.1.2.6.19.2.2.5.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsGatewayVarianceMultiplier	1.3.6.1.4.1.2.6.19.2.2.5.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayDelayAcks	1.3.6.1.4.1.2.6.19.2.2.5.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsRouteTable	1.3.6.1.4.1.2.6.19.2.2.5.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsRouteEntry	1.3.6.1.4.1.2.6.19.2.2.5.2.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRouteDestType	1.3.6.1.4.1.2.6.19.2.2.5.2.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRouteDest	1.3.6.1.4.1.2.6.19.2.2.5.2.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRoutePfxLen	1.3.6.1.4.1.2.6.19.2.2.5.2.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRoutePolicy	1.3.6.1.4.1.2.6.19.2.2.5.2.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRouteNextHopType	1.3.6.1.4.1.2.6.19.2.2.5.2.1.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRouteNextHop	1.3.6.1.4.1.2.6.19.2.2.5.2.1.6	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRouteType	1.3.6.1.4.1.2.6.19.2.2.5.2.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteProto	1.3.6.1.4.1.2.6.19.2.2.5.2.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteAge	1.3.6.1.4.1.2.6.19.2.2.5.2.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteMetric1	1.3.6.1.4.1.2.6.19.2.2.5.2.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteMtu	1.3.6.1.4.1.2.6.19.2.2.5.2.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteReplaceableFlag	1.3.6.1.4.1.2.6.19.2.2.5.2.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteMaximumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.5.2.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteMinimumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.5.2.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteRoundTripGain	1.3.6.1.4.1.2.6.19.2.2.5.2.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteVarianceGain	1.3.6.1.4.1.2.6.19.2.2.5.2.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteVarianceMultiplier	1.3.6.1.4.1.2.6.19.2.2.5.2.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteDelayAcks	1.3.6.1.4.1.2.6.19.2.2.5.2.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRouteFlags	1.3.6.1.4.1.2.6.19.2.2.5.2.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
osasfChannelTable	1.3.6.1.4.1.2.6.19.2.2.6.1	TCP/IP	ibmTCPIPmvsMIB	N/A
osasfChannelEntry	1.3.6.1.4.1.2.6.19.2.2.6.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfChannelNumber	1.3.6.1.4.1.2.6.19.2.2.6.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelType	1.3.6.1.4.1.2.6.19.2.2.6.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelSubType	1.3.6.1.4.1.2.6.19.2.2.6.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelMode	1.3.6.1.4.1.2.6.19.2.2.6.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelHwModel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelState	1.3.6.1.4.1.2.6.19.2.2.6.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelShared	1.3.6.1.4.1.2.6.19.2.2.6.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelNumPorts	1.3.6.1.4.1.2.6.19.2.2.6.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelDeterNodeDesc	1.3.6.1.4.1.2.6.19.2.2.6.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelControlUnitNumber	1.3.6.1.4.1.2.6.19.2.2.6.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCodeLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCurLparName	1.3.6.1.4.1.2.6.19.2.2.6.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCurLparNum	1.3.6.1.4.1.2.6.19.2.2.6.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelManParnName	1.3.6.1.4.1.2.6.19.2.2.6.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelManParnNum	1.3.6.1.4.1.2.6.19.2.2.6.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelFlashLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
osasfPortTable	1.3.6.1.4.1.2.6.19.2.2.6.2	TCP/IP	ibmTCPIPmvsMIB	N/A
osasfPortEntry	1.3.6.1.4.1.2.6.19.2.2.6.2.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfPortNumber	1.3.6.1.4.1.2.6.19.2.2.6.2.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortHardwareState	1.3.6.1.4.1.2.6.19.2.2.6.2.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMediaType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortUniType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortUniVersion	1.3.6.1.4.1.2.6.19.2.2.6.2.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefix	1.3.6.1.4.1.2.6.19.2.2.6.2.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefixPrefix	1.3.6.1.4.1.2.6.19.2.2.6.2.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefixStatus	1.3.6.1.4.1.2.6.19.2.2.6.2.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortCodeLoadStatus	1.3.6.1.4.1.2.6.19.2.2.6.2.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMacAddrBurntIn	1.3.6.1.4.1.2.6.19.2.2.6.2.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMacAddrActive	1.3.6.1.4.1.2.6.19.2.2.6.2.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMaxPcmConnections	1.3.6.1.4.1.2.6.19.2.2.6.2.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortPcmName	1.3.6.1.4.1.2.6.19.2.2.6.2.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortAAL5InPackets	1.3.6.1.4.1.2.6.19.2.2.6.2.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortAAL5OutPackets	1.3.6.1.4.1.2.6.19.2.2.6.2.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortIpAddress	1.3.6.1.4.1.2.6.19.2.2.6.2.1.17	TCP/IP	ibmTCPIPmvsMIB	R/W
osasfPvcTable	1.3.6.1.4.1.2.6.19.2.2.6.3	TCP/IP	ibmTCPIPmvsMIB	N/A
osasfPvcEntry	1.3.6.1.4.1.2.6.19.2.2.6.3.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfPvcName	1.3.6.1.4.1.2.6.19.2.2.6.3.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBestEffort	1.3.6.1.4.1.2.6.19.2.2.6.3.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdPeakCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdPeakCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdsustainCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdsustainCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdCellBurstSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdCellBurstSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcVpi	1.3.6.1.4.1.2.6.19.2.2.6.3.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcVci	1.3.6.1.4.1.2.6.19.2.2.6.3.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdMaxAal5PduSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdMaxAal5PduSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTable	1.3.6.1.4.1.2.6.19.2.2.6.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmSnaLeEntry	1.3.6.1.4.1.2.6.19.2.2.6.4.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmSnaLeLlcTi	1.3.6.1.4.1.2.6.19.2.2.6.4.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeLlcT1	1.3.6.1.4.1.2.6.19.2.2.6.4.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeLlcT2	1.3.6.1.4.1.2.6.19.2.2.6.4.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxStations	1.3.6.1.4.1.2.6.19.2.2.6.4.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxSaps	1.3.6.1.4.1.2.6.19.2.2.6.4.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxIn	1.3.6.1.4.1.2.6.19.2.2.6.4.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsAtmSnaLeMaxOut	1.3.6.1.4.1.2.6.19.2.2.6.4.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeCrsGroupAddress	1.3.6.1.4.1.2.6.19.2.2.6.4.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeCrsUserData	1.3.6.1.4.1.2.6.19.2.2.6.4.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeClientEnableState	1.3.6.1.4.1.2.6.19.2.2.6.4.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeBestEffortPeakRate	1.3.6.1.4.1.2.6.19.2.2.6.4.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxLECConnections	1.3.6.1.4.1.2.6.19.2.2.6.4.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTrEnableLoadBalancing	1.3.6.1.4.1.2.6.19.2.2.6.4.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTrLoadBalancing	1.3.6.1.4.1.2.6.19.2.2.6.4.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTrSessionDelay	1.3.6.1.4.1.2.6.19.2.2.6.4.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigTable	1.3.6.1.4.1.2.6.19.2.2.6.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecConfigEntry	1.3.6.1.4.1.2.6.19.2.2.6.5.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecConfigMode	1.3.6.1.4.1.2.6.19.2.2.6.5.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigLanType	1.3.6.1.4.1.2.6.19.2.2.6.5.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMaxDataFrameSize	1.3.6.1.4.1.2.6.19.2.2.6.5.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigLanName	1.3.6.1.4.1.2.6.19.2.2.6.5.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigLesAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.5.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlTimeout	1.3.6.1.4.1.2.6.19.2.2.6.5.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMaxUnknownFrameCount	1.3.6.1.4.1.2.6.19.2.2.6.5.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecUnknownFrameTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecVccTimeoutPeriod	1.3.6.1.4.1.2.6.19.2.2.6.5.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMaxRetryCount	1.3.6.1.4.1.2.6.19.2.2.6.5.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecAgingTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecForwardDelayTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecExpectedArpResponseTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecFlushTimeout	1.3.6.1.4.1.2.6.19.2.2.6.5.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecPathSwitchingDelay	1.3.6.1.4.1.2.6.19.2.2.6.5.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecLocalSegmentID	1.3.6.1.4.1.2.6.19.2.2.6.5.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendType	1.3.6.1.4.1.2.6.19.2.2.6.5.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendAvgRate	1.3.6.1.4.1.2.6.19.2.2.6.5.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendPeakRate	1.3.6.1.4.1.2.6.19.2.2.6.5.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConnectionCompleteTimer	1.3.6.1.4.1.2.6.19.2.2.6.5.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecPortName	1.3.6.1.4.1.2.6.19.2.2.6.5.1.21	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecStatusTable	1.3.6.1.4.1.2.6.19.2.2.6.6	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecStatusEntry	1.3.6.1.4.1.2.6.19.2.2.6.6.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecPrimaryAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.6.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecID	1.3.6.1.4.1.2.6.19.2.2.6.6.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecInterfaceState	1.3.6.1.4.1.2.6.19.2.2.6.6.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecLastFailureRespCode	1.3.6.1.4.1.2.6.19.2.2.6.6.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecLastFailureState	1.3.6.1.4.1.2.6.19.2.2.6.6.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecProtocol	1.3.6.1.4.1.2.6.19.2.2.6.6.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecVersion	1.3.6.1.4.1.2.6.19.2.2.6.6.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsAtmLecTopologyChange	1.3.6.1.4.1.2.6.19.2.2.6.6.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecconfigServerAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.6.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigSource	1.3.6.1.4.1.2.6.19.2.2.6.6.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecActualLanType	1.3.6.1.4.1.2.6.19.2.2.6.6.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecActualMaxDataFrameSize	1.3.6.1.4.1.2.6.19.2.2.6.6.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecActualLanName	1.3.6.1.4.1.2.6.19.2.2.6.6.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.6.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecProxyClient	1.3.6.1.4.1.2.6.19.2.2.6.6.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecStatisticsTable	1.3.6.1.4.1.2.6.19.2.2.6.7	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecStatisticsEntry	1.3.6.1.4.1.2.6.19.2.2.6.7.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecArpRequestsOut	1.3.6.1.4.1.2.6.19.2.2.6.7.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecArpRequestsIn	1.3.6.1.4.1.2.6.19.2.2.6.7.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecArpRepliesOut	1.3.6.1.4.1.2.6.19.2.2.6.7.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecArpRepliesIn	1.3.6.1.4.1.2.6.19.2.2.6.7.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlFramesOut	1.3.6.1.4.1.2.6.19.2.2.6.7.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlFramesIn	1.3.6.1.4.1.2.6.19.2.2.6.7.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecSvcFailures	1.3.6.1.4.1.2.6.19.2.2.6.7.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecServerTable	1.3.6.1.4.1.2.6.19.2.2.6.8	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecServerEntry	1.3.6.1.4.1.2.6.19.2.2.6.8.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecConfigDirectInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigDirectVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigDirectVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDirectInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDirectVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDirectVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDistributeInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDistributeVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDistributeVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastFwdInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastFwdVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastFwdVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMacAddressTable	1.3.6.1.4.1.2.6.19.2.2.6.9	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecMacAddressEntry	1.3.6.1.4.1.2.6.19.2.2.6.9.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecMacAddress	1.3.6.1.4.1.2.6.19.2.2.6.9.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsTcpConnTable	1.3.6.1.4.1.2.6.19.2.2.7.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsTcpConnEntry	1.3.6.1.4.1.2.6.19.2.2.7.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsTcpConnLastActivity	1.3.6.1.4.1.2.6.19.2.2.7.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnBytesIn	1.3.6.1.4.1.2.6.19.2.2.7.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsTcpConnBytesOut	1.3.6.1.4.1.2.6.19.2.2.7.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOptions	1.3.6.1.4.1.2.6.19.2.2.7.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutBuffered	1.3.6.1.4.1.2.6.19.2.2.7.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnUsrSndNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndUna	1.3.6.1.4.1.2.6.19.2.2.7.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingPush	1.3.6.1.4.1.2.6.19.2.2.7.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingUrg	1.3.6.1.4.1.2.6.19.2.2.7.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingWinSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInBuffered	1.3.6.1.4.1.2.6.19.2.2.7.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRcvNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnUsrRcvNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingPush	1.3.6.1.4.1.2.6.19.2.2.7.1.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingUrg	1.3.6.1.4.1.2.6.19.2.2.7.1.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingWinSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.21	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnReXmt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.22	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnMaxSndWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.23	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnReXmtCount	1.3.6.1.4.1.2.6.19.2.2.7.1.1.24	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnCongestionWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.25	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSSThresh	1.3.6.1.4.1.2.6.19.2.2.7.1.1.26	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRoundTripTime	1.3.6.1.4.1.2.6.19.2.2.7.1.1.27	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRoundTripVariance	1.3.6.1.4.1.2.6.19.2.2.7.1.1.28	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInitSndSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.29	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInitRcvSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.30	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSendMSS	1.3.6.1.4.1.2.6.19.2.2.7.1.1.31	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWl1	1.3.6.1.4.1.2.6.19.2.2.7.1.1.32	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWl2	1.3.6.1.4.1.2.6.19.2.2.7.1.1.33	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.34	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRcvBufSize	1.3.6.1.4.1.2.6.19.2.2.7.1.1.36	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnResourceName	1.3.6.1.4.1.2.6.19.2.2.7.1.1.37	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSubtask	1.3.6.1.4.1.2.6.19.2.2.7.1.1.38	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnResourceId	1.3.6.1.4.1.2.6.19.2.2.7.1.1.39	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSockOpt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.40	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRttSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.44	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTargetAppl	1.3.6.1.4.1.2.6.19.2.2.7.1.1.48	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnLuName	1.3.6.1.4.1.2.6.19.2.2.7.1.1.49	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnClientUserID	1.3.6.1.4.1.2.6.19.2.2.7.1.1.50	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnLogMode	1.3.6.1.4.1.2.6.19.2.2.7.1.1.51	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnProto	1.3.6.1.4.1.2.6.19.2.2.7.1.1.52	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnDupacks	1.3.6.1.4.1.2.6.19.2.2.7.1.1.53	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOptMaxSegmentSize	1.3.6.1.4.1.2.6.19.2.2.7.1.1.54	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsTcpConnClusterConnFlag	1.3.6.1.4.1.2.6.19.2.2.7.1.1.55	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInSegs	1.3.6.1.4.1.2.6.19.2.2.7.1.1.56	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutSegs	1.3.6.1.4.1.2.6.19.2.2.7.1.1.57	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnDSField	1.3.6.1.4.1.2.6.19.2.2.7.1.1.58	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndBufSize	1.3.6.1.4.1.2.6.19.2.2.7.1.1.59	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnAcceptCount	1.3.6.1.4.1.2.6.19.2.2.7.1.1.60	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnExceedBacklog	1.3.6.1.4.1.2.6.19.2.2.7.1.1.61	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnCurrBacklog	1.3.6.1.4.1.2.6.19.2.2.7.1.1.62	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnMaxBacklog	1.3.6.1.4.1.2.6.19.2.2.7.1.1.63	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnWindowScale	1.3.6.1.4.1.2.6.19.2.2.7.1.1.64	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTimeStamp	1.3.6.1.4.1.2.6.19.2.2.7.1.1.65	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnServerResourceId	1.3.6.1.4.1.2.6.19.2.2.7.1.1.66	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnsClosed	1.3.6.1.4.1.2.6.19.2.2.7.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpPassiveDrops	1.3.6.1.4.1.2.6.19.2.2.7.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpTimeWaitReused	1.3.6.1.4.1.2.6.19.2.2.7.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpPredictAck	1.3.6.1.4.1.2.6.19.2.2.7.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpPredictData	1.3.6.1.4.1.2.6.19.2.2.7.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInDupAck	1.3.6.1.4.1.2.6.19.2.2.7.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInBadSum	1.3.6.1.4.1.2.6.19.2.2.7.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInBadLen	1.3.6.1.4.1.2.6.19.2.2.7.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInShort	1.3.6.1.4.1.2.6.19.2.2.7.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInPawsDrop	1.3.6.1.4.1.2.6.19.2.2.7.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInAllBeforeWin	1.3.6.1.4.1.2.6.19.2.2.7.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInSomeBeforeWin	1.3.6.1.4.1.2.6.19.2.2.7.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInAllAfterWin	1.3.6.1.4.1.2.6.19.2.2.7.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInSomeAfterWin	1.3.6.1.4.1.2.6.19.2.2.7.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInOutOfOrder	1.3.6.1.4.1.2.6.19.2.2.7.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInAfterClose	1.3.6.1.4.1.2.6.19.2.2.7.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInWinProbes	1.3.6.1.4.1.2.6.19.2.2.7.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpInWinUpdates	1.3.6.1.4.1.2.6.19.2.2.7.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpOutWinUpdates	1.3.6.1.4.1.2.6.19.2.2.7.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpOutDelayAcks	1.3.6.1.4.1.2.6.19.2.2.7.21	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpOutWinProbes	1.3.6.1.4.1.2.6.19.2.2.7.22	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpRxmtTimers	1.3.6.1.4.1.2.6.19.2.2.7.23	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpRxmtDrops	1.3.6.1.4.1.2.6.19.2.2.7.24	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpPMTURxmts	1.3.6.1.4.1.2.6.19.2.2.7.25	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpPMTUErrors	1.3.6.1.4.1.2.6.19.2.2.7.26	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpProbeDrops	1.3.6.1.4.1.2.6.19.2.2.7.27	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpKeepaliveProbes	1.3.6.1.4.1.2.6.19.2.2.7.28	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpKeepaliveDrops	1.3.6.1.4.1.2.6.19.2.2.7.29	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpFinwait2Drops	1.3.6.1.4.1.2.6.19.2.2.7.30	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmTcpipMvsTcpListenerTable	1.3.6.1.4.1.2.6.19.2.2.7.31	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsTcpListenerEntry	1.3.6.1.4.1.2.6.19.2.2.7.31.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsTcpListenerResourceId	1.3.6.1.4.1.2.6.19.2.2.7.31.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsTcpListenerLocalAddrType	1.3.6.1.4.1.2.6.19.2.2.7.31.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerLocalAddr	1.3.6.1.4.1.2.6.19.2.2.7.31.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerLocalPort	1.3.6.1.4.1.2.6.19.2.2.7.31.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerRemoteAddrType	1.3.6.1.4.1.2.6.19.2.2.7.31.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerRemoteAddr	1.3.6.1.4.1.2.6.19.2.2.7.31.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerRemotePort	1.3.6.1.4.1.2.6.19.2.2.7.31.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerAcceptCount	1.3.6.1.4.1.2.6.19.2.2.7.31.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerExceedBacklog	1.3.6.1.4.1.2.6.19.2.2.7.31.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerCurrBacklog	1.3.6.1.4.1.2.6.19.2.2.7.31.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerMaxBacklog	1.3.6.1.4.1.2.6.19.2.2.7.31.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerResourceName	1.3.6.1.4.1.2.6.19.2.2.7.31.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerCurrConns	1.3.6.1.4.1.2.6.19.2.2.7.31.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerTimeOuts	1.3.6.1.4.1.2.6.19.2.2.7.31.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerAge	1.3.6.1.4.1.2.6.19.2.2.7.31.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsTcpConnectionTable	1.3.6.1.4.1.2.6.19.2.2.7.32	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsTcpConnectionEntry	1.3.6.1.4.1.2.6.19.2.2.7.32.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsTcpConnectionInSegs	1.3.6.1.4.1.2.6.19.2.2.7.32.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionHCInSegs	1.3.6.1.4.1.2.6.19.2.2.7.32.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionOutSegs	1.3.6.1.4.1.2.6.19.2.2.7.32.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionHCOutSegs	1.3.6.1.4.1.2.6.19.2.2.7.32.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionInOctets	1.3.6.1.4.1.2.6.19.2.2.7.32.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionHCInOctets	1.3.6.1.4.1.2.6.19.2.2.7.32.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionOutOctets	1.3.6.1.4.1.2.6.19.2.2.7.32.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionHCOutOctets	1.3.6.1.4.1.2.6.19.2.2.7.32.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionAge	1.3.6.1.4.1.2.6.19.2.2.7.32.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionLastActivity	1.3.6.1.4.1.2.6.19.2.2.7.32.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionResourceName	1.3.6.1.4.1.2.6.19.2.2.7.32.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionResourceId	1.3.6.1.4.1.2.6.19.2.2.7.32.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionSockOpt	1.3.6.1.4.1.2.6.19.2.2.7.32.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionPolicyAction	1.3.6.1.4.1.2.6.19.2.2.7.32.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionPolicyRule	1.3.6.1.4.1.2.6.19.2.2.7.32.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionServerResrcId	1.3.6.1.4.1.2.6.19.2.2.7.32.1.16.	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionApplName	1.3.6.1.4.1.2.6.19.2.2.7.32.1.17.	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionLuName	1.3.6.1.4.1.2.6.19.2.2.7.32.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionLogMode	1.3.6.1.4.1.2.6.19.2.2.7.32.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionProto	1.3.6.1.4.1.2.6.19.2.2.7.32.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionTtlsPolStat	1.3.6.1.4.1.2.6.19.2.2.7.32.1.21	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionTtlsConnStat	1.3.6.1.4.1.2.6.19.2.2.7.32.1.22	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsTcpConnectionTtlsSslProt	1.3.6.1.4.1.2.6.19.2.2.7.32.1.23	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionTtlsNegCipher	1.3.6.1.4.1.2.6.19.2.2.7.32.1.24	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionTtlsSecType	1.3.6.1.4.1.2.6.19.2.2.7.32.1.25	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionTtlsPartUID	1.3.6.1.4.1.2.6.19.2.2.7.32.1.26	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnectionTtlsNegCipher4	1.3.6.1.4.1.2.6.19.2.2.7.32.1.27	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpListenerTableLastChange	1.3.6.1.4.1.2.6.19.2.2.7.33	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpAcceptCount	1.3.6.1.4.1.2.6.19.2.2.7.34	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpHCAcceptCount	1.3.6.1.4.1.2.6.19.2.2.7.35	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsUdpTable	1.3.6.1.4.1.2.6.19.2.2.8.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsUdpEntry	1.3.6.1.4.1.2.6.19.2.2.8.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpLastAct	1.3.6.1.4.1.2.6.19.2.2.8.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpIpOpts	1.3.6.1.4.1.2.6.19.2.2.8.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDgramIn	1.3.6.1.4.1.2.6.19.2.2.8.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpBytesIn	1.3.6.1.4.1.2.6.19.2.2.8.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDgramOut	1.3.6.1.4.1.2.6.19.2.2.8.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpBytesOut	1.3.6.1.4.1.2.6.19.2.2.8.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpResourceName	1.3.6.1.4.1.2.6.19.2.2.8.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSubtask	1.3.6.1.4.1.2.6.19.2.2.8.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpResourceId	1.3.6.1.4.1.2.6.19.2.2.8.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSockOpt	1.3.6.1.4.1.2.6.19.2.2.8.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSendLim	1.3.6.1.4.1.2.6.19.2.2.8.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpRecvLim	1.3.6.1.4.1.2.6.19.2.2.8.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEntryState	1.3.6.1.4.1.2.6.19.2.2.8.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpMcastTTL	1.3.6.1.4.1.2.6.19.2.2.8.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpMcastLoopback	1.3.6.1.4.1.2.6.19.2.2.8.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpMcastLinkAddr	1.3.6.1.4.1.2.6.19.2.2.8.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDSField	1.3.6.1.4.1.2.6.19.2.2.8.1.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpMcastRecvLinkAddr	1.3.6.1.4.1.2.6.19.2.2.8.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsUdpEndpointTable	1.3.6.1.4.1.2.6.19.2.2.8.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsUdpEndpointEntry	1.3.6.1.4.1.2.6.19.2.2.8.3.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpointInDatagrams	1.3.6.1.4.1.2.6.19.2.2.8.3.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointHCInDatagrams	1.3.6.1.4.1.2.6.19.2.2.8.3.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointOutDatagrams	1.3.6.1.4.1.2.6.19.2.2.8.3.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointHCOutDatagrams	1.3.6.1.4.1.2.6.19.2.2.8.3.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointInOctets	1.3.6.1.4.1.2.6.19.2.2.8.3.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointHCInOctets	1.3.6.1.4.1.2.6.19.2.2.8.3.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointOutOctets	1.3.6.1.4.1.2.6.19.2.2.8.3.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointHCOutOctets	1.3.6.1.4.1.2.6.19.2.2.8.3.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointLastActivity	1.3.6.1.4.1.2.6.19.2.2.8.3.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointResourceName	1.3.6.1.4.1.2.6.19.2.2.8.3.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointSockOpt	1.3.6.1.4.1.2.6.19.2.2.8.3.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsUdpEndpointState	1.3.6.1.4.1.2.6.19.2.2.8.3.1.12	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpEndpointMcastHopLim	1.3.6.1.4.1.2.6.19.2.2.8.3.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEndpointMcastIntfName	1.3.6.1.4.1.2.6.19.2.2.8.3.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsUdpEndpMcastTable	1.3.6.1.4.1.2.6.19.2.2.8.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsUdpEndpMcastEntry	1.3.6.1.4.1.2.6.19.2.2.8.4.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastLocalAddrType	1.3.6.1.4.1.2.6.19.2.2.8.4.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastLocalAddr	1.3.6.1.4.1.2.6.19.2.2.8.4.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastLocalPort	1.3.6.1.4.1.2.6.19.2.2.8.4.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastInstance	1.3.6.1.4.1.2.6.19.2.2.8.4.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastRecvAddrType	1.3.6.1.4.1.2.6.19.2.2.8.4.1.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastRecvAddr	1.3.6.1.4.1.2.6.19.2.2.8.4.1.6	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpEndpMcastRecvIntfName	1.3.6.1.4.1.2.6.19.2.2.8.4.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpInDevLayerCalls	1.3.6.1.4.1.2.6.19.2.2.9.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpInUnpackErrors	1.3.6.1.4.1.2.6.19.2.2.9.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpInDiscardsMemory	1.3.6.1.4.1.2.6.19.2.2.9.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpOutDiscardsDlcSynch	1.3.6.1.4.1.2.6.19.2.2.9.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpOutDiscardsDlcAsynch	1.3.6.1.4.1.2.6.19.2.2.9.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsIpOutDiscardsMemory	1.3.6.1.4.1.2.6.19.2.2.9.6	TCP/IP	ibmTCPIPmvsMIB	R/O
osaexpChannelTable	1.3.6.1.4.1.2.6.19.2.2.10.1	TCP/IP	ibmTCPIPmvsMIB	N/A
osaexpChannelEntry	1.3.6.1.4.1.2.6.19.2.2.10.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsOsaExpChannelNumber	1.3.6.1.4.1.2.6.19.2.2.10.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelType	1.3.6.1.4.1.2.6.19.2.2.10.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelSubType	1.3.6.1.4.1.2.6.19.2.2.10.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelMode	1.3.6.1.4.1.2.6.19.2.2.10.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelState	1.3.6.1.4.1.2.6.19.2.2.10.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelShared	1.3.6.1.4.1.2.6.19.2.2.10.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelNumPorts	1.3.6.1.4.1.2.6.19.2.2.10.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelDeterNodeDesc	1.3.6.1.4.1.2.6.19.2.2.10.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelControlUnitNumber	1.3.6.1.4.1.2.6.19.2.2.10.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelCodeLevel	1.3.6.1.4.1.2.6.19.2.2.10.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelCurLparName	1.3.6.1.4.1.2.6.19.2.2.10.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelCurLparNum	1.3.6.1.4.1.2.6.19.2.2.10.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelManLparName	1.3.6.1.4.1.2.6.19.2.2.10.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelManLparNum	1.3.6.1.4.1.2.6.19.2.2.10.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelPCIBusUtil1Min	1.3.6.1.4.1.2.6.19.2.2.10.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelProcessorUtil1Min	1.3.6.1.4.1.2.6.19.2.2.10.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelPCIBusUtil5Min	1.3.6.1.4.1.2.6.19.2.2.10.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelProcessorUtil5Min	1.3.6.1.4.1.2.6.19.2.2.10.1.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelPCIBusUtilHour	1.3.6.1.4.1.2.6.19.2.2.10.1.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpChannelProcessorUtilHour	1.3.6.1.4.1.2.6.19.2.2.10.1.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
osaexpPerfTable	1.3.6.1.4.1.2.6.19.2.2.10.2	TCP/IP	ibmTCPIPmvsMIB	N/A

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
osaexpPerfEntry	1.3.6.1.4.1.2.6.19.2.2.10.2.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsOsaExpPerfLparNum	1.3.6.1.4.1.2.6.19.2.2.10.2.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfProcessorUtil1Min	1.3.6.1.4.1.2.6.19.2.2.10.2.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfInKbytesRate1Min	1.3.6.1.4.1.2.6.19.2.2.10.2.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfOutKbytesRate1Min	1.3.6.1.4.1.2.6.19.2.2.10.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfProcessorUtil5Min	1.3.6.1.4.1.2.6.19.2.2.10.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfInKbytesRate5Min	1.3.6.1.4.1.2.6.19.2.2.10.2.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfOutKbytesRate5Min	1.3.6.1.4.1.2.6.19.2.2.10.2.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfProcessorUtilHour	1.3.6.1.4.1.2.6.19.2.2.10.2.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfInKbytesRateHour	1.3.6.1.4.1.2.6.19.2.2.10.2.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpPerfOutKbytesRateHour	1.3.6.1.4.1.2.6.19.2.2.10.2.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
osaexpEthPortTable	1.3.6.1.4.1.2.6.19.2.2.10.3	TCP/IP	ibmTCPIPmvsMIB	N/A
osaexpEthPortEntry	1.3.6.1.4.1.2.6.19.2.2.10.3.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsOsaExpEthPortNumber	1.3.6.1.4.1.2.6.19.2.2.10.3.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortType	1.3.6.1.4.1.2.6.19.2.2.10.3.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortHardwareState	1.3.6.1.4.1.2.6.19.2.2.10.3.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortServiceMode	1.3.6.1.4.1.2.6.19.2.2.10.3.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortDisabledStatus	1.3.6.1.4.1.2.6.19.2.2.10.3.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortConfigName	1.3.6.1.4.1.2.6.19.2.2.10.3.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortConfigSpeed	1.3.6.1.4.1.2.6.19.2.2.10.3.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortActiveSpeed	1.3.6.1.4.1.2.6.19.2.2.10.3.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortMacAddrActive	1.3.6.1.4.1.2.6.19.2.2.10.3.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortMacAddrBurntIn	1.3.6.1.4.1.2.6.19.2.2.10.3.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortUserData	1.3.6.1.4.1.2.6.19.2.2.10.3.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortOutPackets	1.3.6.1.4.1.2.6.19.2.2.10.3.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortInPackets	1.3.6.1.4.1.2.6.19.2.2.10.3.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortInGroupFrames	1.3.6.1.4.1.2.6.19.2.2.10.3.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortInBroadcastFrames	1.3.6.1.4.1.2.6.19.2.2.10.3.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortName	1.3.6.1.4.1.2.6.19.2.2.10.3.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortInUnknownIPFrames	1.3.6.1.4.1.2.6.19.2.2.10.3.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthPortGroupMacAddrs	1.3.6.1.4.1.2.6.19.2.2.10.3.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
osaexpEthSnaTable	1.3.6.1.4.1.2.6.19.2.2.10.4	TCP/IP	ibmTCPIPmvsMIB	N/A
osaexpEthSnaEntry	1.3.6.1.4.1.2.6.19.2.2.10.4.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsOsaExpEthSnaInactTimer	1.3.6.1.4.1.2.6.19.2.2.10.4.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthSnaRespTimer	1.3.6.1.4.1.2.6.19.2.2.10.4.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthSnaAckTimer	1.3.6.1.4.1.2.6.19.2.2.10.4.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthSnaMaxIFramesBeforeAck	1.3.6.1.4.1.2.6.19.2.2.10.4.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsaExpEthSnaMaxTransmitWindow	1.3.6.1.4.1.2.6.19.2.2.10.4.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPATable	1.3.6.1.4.1.2.6.19.2.2.11.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAEntry	1.3.6.1.4.1.2.6.19.2.2.11.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.1.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsDVIPAIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.1.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAMaskType	1.3.6.1.4.1.2.6.19.2.2.11.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAMaskAddr	1.3.6.1.4.1.2.6.19.2.2.11.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAStatus	1.3.6.1.4.1.2.6.19.2.2.11.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAOrigin	1.3.6.1.4.1.2.6.19.2.2.11.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPARank	1.3.6.1.4.1.2.6.19.2.2.11.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistributeStatus	1.3.6.1.4.1.2.6.19.2.2.11.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAMoveable	1.3.6.1.4.1.2.6.19.2.2.11.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAServMgrEnabled	1.3.6.1.4.1.2.6.19.2.2.11.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAIntfName	1.3.6.1.4.1.2.6.19.2.2.11.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPARangeConfTable	1.3.6.1.4.1.2.6.19.2.2.11.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfEntry	1.3.6.1.4.1.2.6.19.2.2.11.2.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.2.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.2.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfMaskType	1.3.6.1.4.1.2.6.19.2.2.11.2.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfMaskAddr	1.3.6.1.4.1.2.6.19.2.2.11.2.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfMoveable	1.3.6.1.4.1.2.6.19.2.2.11.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPARangeConfStatus	1.3.6.1.4.1.2.6.19.2.2.11.2.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfTable	1.3.6.1.4.1.2.6.19.2.2.11.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfEntry	1.3.6.1.4.1.2.6.19.2.2.11.3.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.3.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.3.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfPort	1.3.6.1.4.1.2.6.19.2.2.11.3.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfTargetDynXcfIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.3.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfTargetDynXcfIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.3.1.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistConfStatus	1.3.6.1.4.1.2.6.19.2.2.11.3.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfTimedAffinity	1.3.6.1.4.1.2.6.19.2.2.11.3.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfSplxPortsEn	1.3.6.1.4.1.2.6.19.2.2.11.3.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfDistMethod	1.3.6.1.4.1.2.6.19.2.2.11.3.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfIntfName	1.3.6.1.4.1.2.6.19.2.2.11.3.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfOptLocal	1.3.6.1.4.1.2.6.19.2.2.11.3.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistConfTargetWeight	1.3.6.1.4.1.2.6.19.2.2.11.3.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAConnRoutingTable	1.3.6.1.4.1.2.6.19.2.2.11.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAConnRoutingEntry	1.3.6.1.4.1.2.6.19.2.2.11.4.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAConnPort	1.3.6.1.4.1.2.6.19.2.2.11.4.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAConnRemIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.4.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAConnRemIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.4.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAConnRemPort	1.3.6.1.4.1.2.6.19.2.2.11.4.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPAConnDynXcfIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.4.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAConnDynXcfIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.4.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAConnPolicyRuleName	1.3.6.1.4.1.2.6.19.2.2.11.4.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsDVIPAConnPolicyActionName	1.3.6.1.4.1.2.6.19.2.2.11.4.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAConnRoute	1.3.6.1.4.1.2.6.19.2.2.11.4.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortTable	1.3.6.1.4.1.2.6.19.2.2.11.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistPortEntry	1.3.6.1.4.1.2.6.19.2.2.11.5.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistPortPort	1.3.6.1.4.1.2.6.19.2.2.11.5.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistPortTargetDynXcfIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.5.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistPortTargetDynXcfIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.5.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPADistPortReadyCount	1.3.6.1.4.1.2.6.19.2.2.11.5.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortTotalConn	1.3.6.1.4.1.2.6.19.2.2.11.5.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortWlmWeight	1.3.6.1.4.1.2.6.19.2.2.11.5.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortDynamicFlag	1.3.6.1.4.1.2.6.19.2.2.11.5.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortFlag	1.3.6.1.4.1.2.6.19.2.2.11.5.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortTsr	1.3.6.1.4.1.2.6.19.2.2.11.5.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortTcsr	1.3.6.1.4.1.2.6.19.2.2.11.5.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortSef	1.3.6.1.4.1.2.6.19.2.2.11.5.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortCer	1.3.6.1.4.1.2.6.19.2.2.11.5.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortAbnormTrans	1.3.6.1.4.1.2.6.19.2.2.11.5.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPADistPortHealth	1.3.6.1.4.1.2.6.19.2.2.11.5.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAServMgrMulticastIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.6.0	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAServMgrMulticastIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.7.0	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAServMgrPort	1.3.6.1.4.1.2.6.19.2.2.11.8.0	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPAServMgrPasswordSpecified	1.3.6.1.4.1.2.6.19.2.2.11.9.0	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPATrapControl	1.3.6.1.4.1.2.6.19.2.2.11.10.0	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsDVIPARangeConfigTable	1.3.6.1.4.1.2.6.19.2.2.11.11	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfigEntry	1.3.6.1.4.1.2.6.19.2.2.11.11.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfigIpAddrType	1.3.6.1.4.1.2.6.19.2.2.11.11.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfigIpAddr	1.3.6.1.4.1.2.6.19.2.2.11.11.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfigPrefixLen	1.3.6.1.4.1.2.6.19.2.2.11.11.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARangeConfigMoveable	1.3.6.1.4.1.2.6.19.2.2.11.11.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPARangeConfigIntfName	1.3.6.1.4.1.2.6.19.2.2.11.11.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPARangeConfigStatus	1.3.6.1.4.1.2.6.19.2.2.11.11.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDVIPARouteTable	1.3.6.1.4.1.2.6.19.2.2.11.12	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARouteEntry	1.3.6.1.4.1.2.6.19.2.2.11.12.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARouteDynXcfType	1.3.6.1.4.1.2.6.19.2.2.11.12.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARouteDynXcfAddr	1.3.6.1.4.1.2.6.19.2.2.11.12.1.2	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARouteTargetType	1.3.6.1.4.1.2.6.19.2.2.11.12.1.3	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARouteTargetAddr	1.3.6.1.4.1.2.6.19.2.2.11.12.1.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsDVIPARouteStatus	1.3.6.1.4.1.2.6.19.2.2.11.12.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTN3270ConnTable	1.3.6.1.4.1.2.6.19.3.1.1.1.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnEntry	1.3.6.1.4.1.2.6.19.3.1.1.1.1.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnLocalAddressType	1.3.6.1.4.1.2.6.19.3.1.1.1.1.1.	TN3270	ibmMvsTN3270MIB	N/A

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsTN3270ConnLocalAddress	1.3.6.1.4.1.2.6.19.3.1.1.1.1.2.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnLocalPort	1.3.6.1.4.1.2.6.19.3.1.1.1.1.3.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnRemAddressType	1.3.6.1.4.1.2.6.19.3.1.1.1.1.4.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnRemAddress	1.3.6.1.4.1.2.6.19.3.1.1.1.1.5.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnRemPort	1.3.6.1.4.1.2.6.19.3.1.1.1.1.6.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270ConnStartTime	1.3.6.1.4.1.2.6.19.3.1.1.1.1.7.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnAppl	1.3.6.1.4.1.2.6.19.3.1.1.1.1.8.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnLuName	1.3.6.1.4.1.2.6.19.3.1.1.1.1.9.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnLogMode	1.3.6.1.4.1.2.6.19.3.1.1.1.1.10.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnProto	1.3.6.1.4.1.2.6.19.3.1.1.1.1.11.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtGroupIndex	1.3.6.1.4.1.2.6.19.3.1.1.1.1.12.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtIpMethod	1.3.6.1.4.1.2.6.19.3.1.1.1.1.13.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtAvgRt	1.3.6.1.4.1.2.6.19.3.1.1.1.1.14.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtAvgIpRt	1.3.6.1.4.1.2.6.19.3.1.1.1.1.15.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtAvgCountTrans	1.3.6.1.4.1.2.6.19.3.1.1.1.1.16.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtIntTimeStamp	1.3.6.1.4.1.2.6.19.3.1.1.1.1.17.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtTotalRts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.18.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtTotalIpRts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.19.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtCountTrans	1.3.6.1.4.1.2.6.19.3.1.1.1.1.20.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtCountIP	1.3.6.1.4.1.2.6.19.3.1.1.1.1.21.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtElapsRndTrpSq	1.3.6.1.4.1.2.6.19.3.1.1.1.1.22.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtElapsIpRtSq	1.3.6.1.4.1.2.6.19.3.1.1.1.1.23.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtElapsSnaRtSq	1.3.6.1.4.1.2.6.19.3.1.1.1.1.24.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtBucket1Rts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.25.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtBucket2Rts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.26.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtBucket3Rts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.27.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtBucket4Rts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.28.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270ConnRtBucket5Rts	1.3.6.1.4.1.2.6.19.3.1.1.1.1.29.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupTable	1.3.6.1.4.1.2.6.19.3.1.2.1.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270MonGroupEntry	1.3.6.1.4.1.2.6.19.3.1.2.1.1.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270MonGroupIndex	1.3.6.1.4.1.2.6.19.3.1.2.1.1.1.	TN3270	ibmMvsTN3270MIB	N/A
ibmMvsTN3270MonGroupName	1.3.6.1.4.1.2.6.19.3.1.2.1.1.2.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupType	1.3.6.1.4.1.2.6.19.3.1.2.1.1.3.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupSampPeriod	1.3.6.1.4.1.2.6.19.3.1.2.1.1.4.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupSampMult	1.3.6.1.4.1.2.6.19.3.1.2.1.1.5.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupBucketBndry1	1.3.6.1.4.1.2.6.19.3.1.2.1.1.6.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupBucketBndry2	1.3.6.1.4.1.2.6.19.3.1.2.1.1.7.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupBucketBndry3	1.3.6.1.4.1.2.6.19.3.1.2.1.1.8.	TN3270	ibmMvsTN3270MIB	R/O
ibmMvsTN3270MonGroupBucketBndry4	1.3.6.1.4.1.2.6.19.3.1.2.1.1.9.	TN3270	ibmMvsTN3270MIB	R/O
snmpSetSerialNo	1.3.6.1.6.3.1.1.6.1	Agent	RFC1907	R/O
snmpEngineID	1.3.6.1.6.3.10.2.1.1	Agent	RFC2571	R/O

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
snmpEngineBoots	1.3.6.1.6.3.10.2.1.2	Agent	RFC2571	R/O
snmpEngineTime	1.3.6.1.6.3.10.2.1.3	Agent	RFC2571	R/O
snmpEngineMaxMessageSize	1.3.6.1.6.3.10.2.1.4	Agent	RFC2571	R/O
snmpUnknownSecurityModels	1.3.6.1.6.3.11.2.1.1	Agent	RFC2572	R/O
snmpInvalidMsgs	1.3.6.1.6.3.11.2.1.2	Agent	RFC2572	R/O
snmpUnknownPDUHandlers	1.3.6.1.6.3.11.2.1.3	Agent	RFC2572	R/O
snmpTargetSpinLock	1.3.6.1.6.3.12.1.1	Agent	RFC2573	R/W
snmpTargetAddrTable	1.3.6.1.6.3.12.1.2	Agent	RFC2573	N/A
snmpTargetAddrEntry	1.3.6.1.6.3.12.1.2.1	Agent	RFC2573	N/A
snmpTargetAddrName	1.3.6.1.6.3.12.1.2.1.1	Agent	RFC2573	N/A
snmpTargetAddrTDomain	1.3.6.1.6.3.12.1.2.1.2	Agent	RFC2573	R/C
snmpTargetAddrTAddress	1.3.6.1.6.3.12.1.2.1.3	Agent	RFC2573	R/C
snmpTargetAddrTimeout	1.3.6.1.6.3.12.1.2.1.4	Agent	RFC2573	R/C
snmpTargetAddrRetryCount	1.3.6.1.6.3.12.1.2.1.5	Agent	RFC2573	R/C
snmpTargetAddrTagList	1.3.6.1.6.3.12.1.2.1.6	Agent	RFC2573	R/C
snmpTargetAddrParams	1.3.6.1.6.3.12.1.2.1.7	Agent	RFC2573	R/C
snmpTargetAddrStorageType	1.3.6.1.6.3.12.1.2.1.8	Agent	RFC2573	R/C
snmpTargetAddrRowStatus	1.3.6.1.6.3.12.1.2.1.9	Agent	RFC2573	R/C
snmpTargetParamsTable	1.3.6.1.6.3.12.1.3	Agent	RFC2573	N/A
snmpTargetParamsEntry	1.3.6.1.6.3.12.1.3.1	Agent	RFC2573	N/A
snmpTargetParamsName	1.3.6.1.6.3.12.1.3.1.1	Agent	RFC2573	N/A
snmpTargetParamsMPModel	1.3.6.1.6.3.12.1.3.1.2	Agent	RFC2573	R/C
snmpTargetParamsSecurityModel	1.3.6.1.6.3.12.1.3.1.3	Agent	RFC2573	R/C
snmpTargetParamsSecurityName	1.3.6.1.6.3.12.1.3.1.4	Agent	RFC2573	R/C
snmpTargetParamsSecurityLevel	1.3.6.1.6.3.12.1.3.1.5	Agent	RFC2573	R/C
snmpTargetParamsStorageType	1.3.6.1.6.3.12.1.3.1.6	Agent	RFC2573	R/C
snmpTargetParamsRowStatus	1.3.6.1.6.3.12.1.3.1.7	Agent	RFC2573	R/C
snmpUnavailableContexts	1.3.6.1.6.3.12.1.4	Agent	RFC2573	R/O
snmpUnknownContexts	1.3.6.1.6.3.12.1.5	Agent	RFC2573	R/O
snmpNotifyTable	1.3.6.1.6.3.13.1.1	Agent	RFC2573	N/A
snmpNotifyEntry	1.3.6.1.6.3.13.1.1.1	Agent	RFC2573	N/A
snmpNotifyName	1.3.6.1.6.3.13.1.1.1.1	Agent	RFC2573	N/A
snmpNotifyTag	1.3.6.1.6.3.13.1.1.1.2	Agent	RFC2573	R/C
snmpNotifyType	1.3.6.1.6.3.13.1.1.1.3	Agent	RFC2573	R/C
snmpNotifyStorageType	1.3.6.1.6.3.13.1.1.1.4	Agent	RFC2573	R/C
snmpNotifyRowStatus	1.3.6.1.6.3.13.1.1.1.5	Agent	RFC2573	R/C
snmpNotifyFilterProfileTable	1.3.6.1.6.3.13.1.2	Agent	RFC2573	N/A
snmpNotifyFilterProfileEntry	1.3.6.1.6.3.13.1.2.1	Agent	RFC2573	N/A
snmpNotifyFilterProfileName	1.3.6.1.6.3.13.1.2.1.1	Agent	RFC2573	R/C
snmpNotifyFilterProfileStorType	1.3.6.1.6.3.13.1.2.1.2	Agent	RFC2573	R/C
snmpNotifyFilterProfileRowStatus	1.3.6.1.6.3.13.1.2.1.3	Agent	RFC2573	R/C

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
snmpNotifyFilterTable	1.3.6.1.6.3.13.1.3	Agent	RFC2573	N/A
snmpNotifyFilterEntry	1.3.6.1.6.3.13.1.3.1	Agent	RFC2573	N/A
snmpNotifyFilterSubtree	1.3.6.1.6.3.13.1.3.1.1	Agent	RFC2573	N/A
snmpNotifyFilterMask	1.3.6.1.6.3.13.1.3.1.2	Agent	RFC2573	R/C
snmpNotifyFilterType	1.3.6.1.6.3.13.1.3.1.3	Agent	RFC2573	R/C
snmpNotifyFilterStorageType	1.3.6.1.6.3.13.1.3.1.4	Agent	RFC2573	R/C
snmpNotifyFilterRowStatus	1.3.6.1.6.3.13.1.3.1.5	Agent	RFC2573	R/C
usmStatsUnsupportedSecLevels	1.3.6.1.6.3.15.1.1.1	Agent	RFC2574	R/O
usmStatsNotInTimeWindows	1.3.6.1.6.3.15.1.1.2	Agent	RFC2574	R/O
usmStatsUnknownUserNames	1.3.6.1.6.3.15.1.1.3	Agent	RFC2574	R/O
usmStatsUnknownEngineIDs	1.3.6.1.6.3.15.1.1.4	Agent	RFC2574	R/O
usmStatsWrongDigests	1.3.6.1.6.3.15.1.1.5	Agent	RFC2574	R/O
usmStatsDecryptionErrors	1.3.6.1.6.3.15.1.1.6	Agent	RFC2574	R/O
usmUserSpinLock	1.3.6.1.6.3.15.1.2.1	Agent	RFC2574	R/W
usmUserTable	1.3.6.1.6.3.15.1.2.2	Agent	RFC2574	N/A
usmUserEntry	1.3.6.1.6.3.15.1.2.2.1	Agent	RFC2574	N/A
usmUserEngineID	1.3.6.1.6.3.15.1.2.2.1.1	Agent	RFC2574	N/A
usmUserName	1.3.6.1.6.3.15.1.2.2.1.2	Agent	RFC2574	N/A
usmUserSecurityName	1.3.6.1.6.3.15.1.2.2.1.3	Agent	RFC2574	R/O
usmUserCloneFrom	1.3.6.1.6.3.15.1.2.2.1.4	Agent	RFC2574	R/C
usmUserAuthProtocol	1.3.6.1.6.3.15.1.2.2.1.5	Agent	RFC2574	R/C
usmUserAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.6	Agent	RFC2574	R/C
usmUserOwnAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.7	Agent	RFC2574	R/C
usmUserPrivProtocol	1.3.6.1.6.3.15.1.2.2.1.8	Agent	RFC2574	R/C
usmUserPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.9	Agent	RFC2574	R/C
usmUserOwnPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.10	Agent	RFC2574	R/C
usmUserPublic	1.3.6.1.6.3.15.1.2.2.1.11	Agent	RFC2574	R/C
usmUserStorageType	1.3.6.1.6.3.15.1.2.2.1.12	Agent	RFC2574	R/C
usmUserStatus	1.3.6.1.6.3.15.1.2.2.1.13	Agent	RFC2574	R/C
vacmContextTable	1.3.6.1.6.3.16.1.1	Agent	RFC2575	N/A
vacmContextEntry	1.3.6.1.6.3.16.1.1.1	Agent	RFC2575	N/A
vacmContextName	1.3.6.1.6.3.16.1.1.1.1	Agent	RFC2575	R/O
vacmSecurityToGroupTable	1.3.6.1.6.3.16.1.2	Agent	RFC2575	N/A
vacmSecurityToGroupEntry	1.3.6.1.6.3.16.1.2.1	Agent	RFC2575	N/A
vacmSecurityModel	1.3.6.1.6.3.16.1.2.1.1	Agent	RFC2575	N/A
vacmSecurityName	1.3.6.1.6.3.16.1.2.1.2	Agent	RFC2575	N/A
vacmGroupName	1.3.6.1.6.3.16.1.2.1.3	Agent	RFC2575	R/C
vacmSecurityToGroupStorageType	1.3.6.1.6.3.16.1.2.1.4	Agent	RFC2575	R/C
vacmSecurityToGroupStatus	1.3.6.1.6.3.16.1.2.1.5	Agent	RFC2575	R/C
vacmAccessTable	1.3.6.1.6.3.16.1.4	Agent	RFC2575	N/A
vacmAccessEntry	1.3.6.1.6.3.16.1.4.1	Agent	RFC2575	N/A

Table 22. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
vacmAccessContextPrefix	1.3.6.1.6.3.16.1.4.1.1	Agent	RFC2575	N/A
vacmAccessSecurityModel	1.3.6.1.6.3.16.1.4.1.2	Agent	RFC2575	N/A
vacmAccessSecurityLevel	1.3.6.1.6.3.16.1.4.1.3	Agent	RFC2575	N/A
vacmAccessContextMatch	1.3.6.1.6.3.16.1.4.1.4	Agent	RFC2575	R/C
vacmAccessReadViewName	1.3.6.1.6.3.16.1.4.1.5	Agent	RFC2575	R/C
vacmAccessWriteViewName	1.3.6.1.6.3.16.1.4.1.6	Agent	RFC2575	R/C
vacmAccessNotifyViewName	1.3.6.1.6.3.16.1.4.1.7	Agent	RFC2575	R/C
vacmAccessStorageType	1.3.6.1.6.3.16.1.4.1.8	Agent	RFC2575	R/C
vacmAccessStatus	1.3.6.1.6.3.16.1.4.1.9	Agent	RFC2575	R/C
vacmViewSpinLock	1.3.6.1.6.3.16.1.5.1	Agent	RFC2575	R/W
vacmViewTreeFamilyTable	1.3.6.1.6.3.16.1.5.2	Agent	RFC2575	N/A
vacmViewTreeFamilyEntry	1.3.6.1.6.3.16.1.5.2.1	Agent	RFC2575	N/A
vacmViewTreeFamilyViewName	1.3.6.1.6.3.16.1.5.2.1.1	Agent	RFC2575	N/A
vacmViewTreeFamilySubtree	1.3.6.1.6.3.16.1.5.2.1.2	Agent	RFC2575	N/A
vacmViewTreeFamilyMask	1.3.6.1.6.3.16.1.5.2.1.3	Agent	RFC2575	R/C
vacmViewTreeFamilyType	1.3.6.1.6.3.16.1.5.2.1.4	Agent	RFC2575	R/C
vacmViewTreeFamilyStorageType	1.3.6.1.6.3.16.1.5.2.1.5	Agent	RFC2575	R/C
vacmViewTreeFamilyStatus	1.3.6.1.6.3.16.1.5.2.1.6	Agent	RFC2575	R/C
snmpCommunityTable	1.3.6.1.6.3.18.1.1	Agent	RFC2576	N/A
snmpCommunityEntry	1.3.6.1.6.3.18.1.1.1	Agent	RFC2576	N/A
snmpCommunityIndex	1.3.6.1.6.3.18.1.1.1.1	Agent	RFC2576	N/A
snmpCommunityName	1.3.6.1.6.3.18.1.1.1.2	Agent	RFC2576	R/C
snmpCommunitySecurityName	1.3.6.1.6.3.18.1.1.1.3	Agent	RFC2576	R/C
snmpCommunityContextEngineID	1.3.6.1.6.3.18.1.1.1.4	Agent	RFC2576	R/C
snmpCommunityContextName	1.3.6.1.6.3.18.1.1.1.5	Agent	RFC2576	R/C
snmpCommunityTransportTag	1.3.6.1.6.3.18.1.1.1.6	Agent	RFC2576	R/C
snmpCommunityStorageType	1.3.6.1.6.3.18.1.1.1.7	Agent	RFC2576	R/C
snmpCommunityStatus	1.3.6.1.6.3.18.1.1.1.8	Agent	RFC2576	R/C
snmpTargetAddrExtTable	1.3.6.1.6.3.18.1.2	Agent	RFC2576	N/A
snmpTargetAddrExtEntry	1.3.6.1.6.3.18.1.2.1	Agent	RFC2576	N/A
snmpTargetAddrTMask	1.3.6.1.6.3.18.1.2.1.1	Agent	RFC2576	R/C
snmpTargetAddr MMS	1.3.6.1.6.3.18.1.2.1.2	Agent	RFC2576	R/C

Appendix C. IBM 3172 attribute index

This topic shows the 3172 attributes and their corresponding MIB variables.

Table 23. MIB variable cross-reference table

3172 attribute	MIB variable
01	= ibm3172Descr
02	= ibm3172Contact
03	= ibm3172Location
04	= ibm3172ifNumber
10	= ibm3172ifTrapEnable
11	= ifDescr
12	= ifType
13	= ifPhysAddress
14	= ifOperStatus
20	= ibm3172ifChanCounters
21	= ibm3172ifInChanOctets
22	= ibm3172ifOutChanOctets
23	= ibm3172ifInChanBlocks
24	= ibm3172ifOutChanBlocks
30	= ibm3172ifLANCounters
31	= ibm3172ifInLANOctets
32	= ibm3172ifOutLANOctets
33	= ibm3172ifInLANFrames
34	= ibm3172ifOutLANFrames
35	= ibm3172ifInLANErrors
36	= ibm3172ifOutLANErrors
37	= ibm3172ifInLANDiscards
38	= ibm3172ifOutLANDiscards
40	= ibm3172ifBlkCounters
41	= ibm3172ifBlkRcvOctets
42	= ibm3172ifBlkXmitOctets
43	= ibm3172ifBlkRcvFrames
44	= ibm3172ifBlkXmitBlocks
45	= ibm3172ifInBlkErrors
46	= ibm3172ifInBlkDiscards
50	= ibm3172ifDblkCounters
51	= ibm3172ifDblkRcvOctets
52	= ibm3172ifDblkXmitOctets
53	= ibm3172ifDblkRcvBlocks
54	= ibm3172ifDblkXmitFrames

Table 23. MIB variable cross-reference table (continued)

3172 attribute	MIB variable
55	= ibm3172ifOutDblkErrors
56	= ibm3172ifOutDblkDiscards

Appendix D. SNMP trap types

This topic lists the generic and Enterprise-specific trap types that can be received by SNMP.

SNMP Generic trap types

Table 24 lists the generic trap types that can be received by SNMP.

Table 24. Generic trap types

Value	Type	Description
0	coldStart	A coldStart trap signifies that the sending protocol entity is reinitializing itself so that the agent's configuration or the protocol entity implementation can be altered.
1	warmStart	A warmStart trap signifies that the sending protocol entity is reinitializing itself so that neither the agent configuration nor the protocol entity implementation can be altered.
2	linkDown	A linkDown trap signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration. A Trap-PDU of type linkDown contains, as the first element of its variable-bindings, the name and value of the ifIndex instance for the affected interface.
3	linkUp	A linkUp trap signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up. A Trap-PDU of type linkUp contains, as the first element of its variable-bindings, the name and value of the ifIndex instance for the affected interface.
4	authenticationFailure	An authenticationFailure trap signifies that the sending protocol entity is the addressee of a protocol message that is not properly authenticated.
5	egpNeighborLoss	An egpNeighborLoss trap signifies that an EGP neighbor for whom the sending protocol entity was an EGP peer has been marked down and the peer relationship no longer exists. The Trap-PDU of the egpNeighborLoss contains, as the first element of its variable-bindings, the name and value of the egpNeighAddr instance for the affected neighbor.
6	enterpriseSpecific	An enterpriseSpecific trap signifies that the sending protocol entity recognizes that some Enterprise-specific event has occurred. The specific-trap field identifies the particular trap that occurred.

SNMP Enterprise-specific trap types

Table 25 lists the Enterprise-specific trap types generated by subagents shipped with z/OS Communications Server. All Enterprise-specific traps are generated with the trap value 6.

Table 25. MVS Enterprise trap types

Value	Subagent	Type	Description
1	TCP/IP	ibmMvsAtmOsasfAtmPvcCreate	This trap is generated when OSA/SF sends an asyn notification to the TCP/IP DPI Subagent that a PVC was created for a given OSA-2 ATM. This notification contains the corresponding ibmMvsAtmOsasfPortName instance. Representation of this contains the port's (aal5 layer interface) 'ifIndex.pvcNameOctetCount. pvcNameInASCIINvt'.
2	TCP/IP	ibmMvsAtmOsasfAtmPvcDelete	This trap is generated when OSA/SF sends an asyn notification to the TCP/IP DPI Subagent that a PVC was deleted for a given OSA-2 ATM. This notification contains the corresponding ibmMvsAtmOsasfPortName instance. Representation of this contains the port's (aal5 layer interface) 'ifIndex.pvcNameOctetCount. pvcNameInASCIINvt'.
3	TCP/IP	ibmMvsDVIPAStatusChange	This trap is generated when a dynamic VIPA interface is either defined to a TCP/IP stack or its status changes. This notification contains the status, the origin value, the rank value, the moveable attribute, and the service manager indicator. The origin value indicates why the dynamic VIPA interface was originally defined.
4	TCP/IP	ibmMvsDVIPARemoved	This trap is generated when a dynamic VIPA interface is removed from a TCP/IP stack. This notification contains the status, the origin value, the rank value, the moveable attribute, and the service manager indicator prior to removal. The origin value indicates why the dynamic VIPA interface was previously activated.
5	TCP/IP	ibmMvsDVIPATargetAdded	This trap is generated by a sysplex distributor stack when it determines a designated target stack is active. Stacks are designated as target stacks on the VIPADISTRIBUTE profile statement. This notification contains the ibmMvsDVIPADistConfStatus object whose instance indicates the dynamic VIPA IP address, distributed port, and target stack dynamic XCF IP address.

Table 25. MVS Enterprise trap types (continued)

Value	Subagent	Type	Description
6	TCP/IP	ibmMvsDVIPATargetRemoved	This trap is by a sysplex distributor stack when an active target stack is removed from distribution. This can occur when a VIPADISTRIBUTE DELETE profile statement is processed, or the target stack ends. This notification contains the ibmMvsDVIPADistConfStatus object whose instance indicates the dynamic VIPA IP address, distributed port, and target stack dynamic XCF IP address.
7	TCP/IP	ibmMvsDVIPATargetServerStarted	This trap is generated by a sysplex distributor stack when it receives notification from a target stack that a server has become active on a distributed port. This notification contains the count of servers ready at the port and the instance indicates the dynamic VIPA IP address, the distributed port, and the target stack dynamic XCF IP address.
8	TCP/IP	ibmMvsDVIPATargetServerEnded	This trap is generated by a sysplex distributor stack when it receives notification from a target stack that a server has ended on a distributed port. This notification contains the count of servers ready at the port and the instance indicates the dynamic VIPA IP address, the distributed port, and the target stack dynamic XCF IP address.
9	TCP/IP	ibmMvsTcpipSubagentColdStart	This trap is generated by the TCP/IP Subagent. It signifies that the Subagent, acting in a subagent role, has reintialized itself and that its configuration might have been altered.
10	TCP/IP	ibmMvsTcpipIntfDown	This trap is generated when a network interface transitions to the down state, meaning that is it now inactive. This trap is similar to the standard linkDown trap but provides the ifName value of the interface, along with the interface index and status values.
11	TCP/IP	ibmMvsTcpipIntfUp	This trap is generated when a network interface transitions to the up state, meaning that is it now active. This trap is similar to the standard linkUp trap but provides the ifName value of the interface, along with the interface index and status values.

Table 25. MVS Enterprise trap types (continued)

Value	Subagent	Type	Description
1	Network SLAPM2	slapm2PolicyRuleMonNotOkay	This notification is generated when one or more of the following three monitored quantities goes above its high threshold, indicating that its value has become unacceptable: • slapm2PRMonTcpRttCurrentDelay • slapm2PRMonCurrentTcpReXmit • slapm2PRMonAcceptQCurrentDelay The first slapm2PRMonStatus value supplies the current monitor statuses for these three quantities, and the second value supplies the previous values. For a rising quantity, the bit in the previous status is set to off, indicating that the quantity is below the high threshold, and the bit in the current status is set to on, indicating that the quantity is above the high threshold. By examining these two values, it is possible to determine which monitored quantity (or quantities) caused the notification to be issued. slapm2PRMonTrapEnable for the conceptual row must be set to enabled for this notification to be generated. Also, see the definitions of the high threshold objects for a description of the hysteresis behavior for this notification, which reduces the number of notifications that are generated when reporting is enabled.

Table 25. MVS Enterprise trap types (continued)

Value	Subagent	Type	Description
2	Network SLAPM2	slapm2PolicyRuleMonOkay	This notification is generated when one or more of the following three monitored quantities goes below its low threshold, indicating that its value returned to an acceptable level: • slapm2PRMonTcpRttCurrentDelay • slapm2PRMonCurrentTcpReXmit • slapm2PRMonAcceptQCurrentDelay The first slapm2PRMonStatus value supplies the current monitor statuses for these three quantities, and the second value supplies their previous values. For a falling quantity, the bit in the previous status is set to on, indicating that the quantity is above the low threshold, and the bit in the current status is set to off, indicating that the quantity is below the low threshold. By examining these two values, it is possible to determine which monitored quantity (or quantities) caused the notification to be issued. slapm2PRMonTrapEnable for the conceptual row must be set to enabled for this notification to be generated. Also, see the definitions of the low threshold objects for a description of the hysteresis behavior for this notification, which reduces the number of notifications that are generated when reporting is enabled.
3	Network SLAPM2	slapm2PolicyRuleDeleted	A slapm2PolicyRuleDeleted notification is sent when a slapm2PolicyRuleStatsEntry is deleted if the value of slapm2PolicyTrapDeletedEnable is enabled(1).
4	Nework SLAPM2	slapm2PolicyRuleMonDeleted	A slapm2PolicyRuleMonDeleted notification is sent when a slapm2PRMonEntry is deleted if the value of slapm2PolicyDeletedTrapEnable is enabled(1).

Appendix E. ICMP/ICMPv6 types and codes

For information about the Internet Control Message Protocol (ICMP) types and codes, see http://www.iana.org/assignments/icmp-parameters.

For information about the Internet Control Message Protocol for IPv6 (ICMPv6) types and codes, see http://www.iana.org/assignments/icmpv6-parameters.

Appendix F. Related protocol specifications

This appendix lists the related protocol specifications (RFCs) for TCP/IP. The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. Some of these protocols are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement these particular functions or protocols. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

You can request RFCs through electronic mail, from the automated Network Information Center (NIC) mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC` *nnnn* for text versions or a subject line of `RFC` *nnnn*`.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil` or at:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Hard copies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available at the following Web address: http://www.rfc-editor.org/rfc.html.

Draft RFCs that have been implemented in this and previous Communications Server releases are listed at the end of this topic.

Many features of TCP/IP Services are based on the following RFCs:

RFC **Title and Author**

RFC 652
Telnet output carriage-return disposition option D. Crocker

RFC 653
Telnet output horizontal tabstops option D. Crocker

RFC 654
Telnet output horizontal tab disposition option D. Crocker

RFC 655
Telnet output formfeed disposition option D. Crocker

RFC 657
Telnet output vertical tab disposition option D. Crocker

RFC 658
Telnet output linefeed disposition D. Crocker

RFC 698
Telnet extended ASCII option T. Mock

RFC 726
Remote Controlled Transmission and Echoing Telnet option J. Postel, D. Crocker

RFC 727
Telnet logout option M.R. Crispin

RFC 732
Telnet Data Entry Terminal option J.D. Day

RFC 733
Standard for the format of ARPA network text messages D. Crocker, J. Vittal, K.T. Pogran, D.A. Henderson

RFC 734
SUPDUP Protocol M.R. Crispin

RFC 735
Revised Telnet byte macro option D. Crocker, R.H. Gumpertz

RFC 736
Telnet SUPDUP option M.R. Crispin

RFC 749
Telnet SUPDUP—Output option B. Greenberg

RFC 765
File Transfer Protocol specification J. Postel

RFC 768
User Datagram Protocol J. Postel

RFC 779
Telnet send-location option E. Killian

RFC 783
TFTP Protocol (revision 2) K.R. Sollins

RFC 791
Internet Protocol J. Postel

RFC 792
Internet Control Message Protocol J. Postel

RFC 793
Transmission Control Protocol J. Postel

RFC 820
Assigned numbers J. Postel

RFC 821
Simple Mail Transfer Protocol J. Postel

RFC 822
Standard for the format of ARPA Internet text messages D. Crocker

RFC 823
DARPA Internet gateway R. Hinden, A. Sheltzer

RFC 826
Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware D. Plummer

RFC 854
Telnet Protocol Specification J. Postel, J. Reynolds

RFC 855
Telnet Option Specification J. Postel, J. Reynolds

RFC 856
Telnet Binary Transmission J. Postel, J. Reynolds

RFC 857
Telnet Echo Option J. Postel, J. Reynolds

RFC 858
Telnet Suppress Go Ahead Option J. Postel, J. Reynolds

RFC 859
Telnet Status Option J. Postel, J. Reynolds

RFC 860
Telnet Timing Mark Option J. Postel, J. Reynolds

RFC 861
Telnet Extended Options: List Option J. Postel, J. Reynolds

RFC 862
Echo Protocol J. Postel

RFC 863
Discard Protocol J. Postel

RFC 864
Character Generator Protocol J. Postel

RFC 865
Quote of the Day Protocol J. Postel

RFC 868
Time Protocol J. Postel, K. Harrenstien

RFC 877
Standard for the transmission of IP datagrams over public data networks J.T. Korb

RFC 883
Domain names: Implementation specification P.V. Mockapetris

RFC 884
Telnet terminal type option M. Solomon, E. Wimmers

RFC 885
Telnet end of record option J. Postel

RFC 894
Standard for the transmission of IP datagrams over Ethernet networks C. Hornig

RFC 896
Congestion control in IP/TCP internetworks J. Nagle

RFC 903
Reverse Address Resolution Protocol R. Finlayson, T. Mann, J. Mogul, M. Theimer

RFC 904
Exterior Gateway Protocol formal specification D. Mills

RFC 919
Broadcasting Internet Datagrams J. Mogul

RFC 922
Broadcasting Internet datagrams in the presence of subnets J. Mogul

RFC 927
TACACS user identification Telnet option B.A. Anderson

RFC 933
Output marking Telnet option S. Silverman

RFC 946
Telnet terminal location number option R. Nedved

RFC 950
Internet Standard Subnetting Procedure J. Mogul, J. Postel

RFC 952
DoD Internet host table specification K. Harrenstien, M. Stahl, E. Feinler

RFC 959
File Transfer Protocol J. Postel, J.K. Reynolds

RFC 961
Official ARPA-Internet protocols J.K. Reynolds, J. Postel

RFC 974
Mail routing and the domain system C. Partridge

RFC 1001
Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force

RFC 1002
Protocol Standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force

RFC 1006
ISO transport services on top of the TCP: Version 3 M.T. Rose, D.E. Cass

RFC 1009
Requirements for Internet gateways R. Braden, J. Postel

RFC 1011
Official Internet protocols J. Reynolds, J. Postel

RFC 1013
X Window System Protocol, version 11: Alpha update April 1987 R. Scheifler

RFC 1014
XDR: External Data Representation standard Sun Microsystems

RFC 1027
Using ARP to implement transparent subnet gateways S. Carl-Mitchell, J. Quarterman

RFC 1032
Domain administrators guide M. Stahl

RFC 1033
Domain administrators operations guide M. Lottor

RFC 1034
Domain names—concepts and facilities P.V. Mockapetris

RFC 1035
Domain names—implementation and specification P.V. Mockapetris

RFC 1038
Draft revised IP security option M. St. Johns

RFC 1041
Telnet 3270 regime option Y. Rekhter

RFC 1042
Standard for the transmission of IP datagrams over IEEE 802 networks J. Postel, J. Reynolds

RFC 1043
Telnet Data Entry Terminal option: DODIIS implementation A. Yasuda, T. Thompson

RFC 1044
Internet Protocol on Network System's HYPERchannel: Protocol specification K. Hardwick, J. Lekashman

RFC 1053
Telnet X.3 PAD option S. Levy, T. Jacobson

RFC 1055
Nonstandard for transmission of IP datagrams over serial lines: SLIP J. Romkey

RFC 1057
RPC: Remote Procedure Call Protocol Specification: Version 2 Sun Microsystems

RFC 1058
Routing Information Protocol C. Hedrick

RFC 1060
Assigned numbers J. Reynolds, J. Postel

RFC 1067
Simple Network Management Protocol J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin

RFC 1071
Computing the Internet checksum R.T. Braden, D.A. Borman, C. Partridge

RFC 1072
TCP extensions for long-delay paths V. Jacobson, R.T. Braden

RFC 1073
Telnet window size option D. Waitzman

RFC 1079
Telnet terminal speed option C. Hedrick

RFC 1085
ISO presentation services on top of TCP/IP based internets M.T. Rose

RFC 1091
Telnet terminal-type option J. VanBokkelen

RFC 1094
NFS: Network File System Protocol specification Sun Microsystems

RFC 1096
Telnet X display location option G. Marcy

RFC 1101
DNS encoding of network names and other types P. Mockapetris

RFC 1112
Host extensions for IP multicasting S.E. Deering

RFC 1113
Privacy enhancement for Internet electronic mail: Part I — message encipherment and authentication procedures J. Linn

RFC 1118
Hitchhikers Guide to the Internet E. Krol

RFC 1122
Requirements for Internet Hosts—Communication Layers R. Braden, Ed.

RFC 1123
Requirements for Internet Hosts—Application and Support R. Braden, Ed.

RFC 1146
TCP alternate checksum options J. Zweig, C. Partridge

RFC 1155
Structure and identification of management information for TCP/IP-based internets M. Rose, K. McCloghrie

RFC 1156
Management Information Base for network management of TCP/IP-based internets K. McCloghrie, M. Rose

RFC 1157
Simple Network Management Protocol (SNMP) J. Case, M. Fedor, M. Schoffstall, J. Davin

RFC 1158
Management Information Base for network management of TCP/IP-based internets: MIB-II M. Rose

RFC 1166
Internet numbers S. Kirkpatrick, M.K. Stahl, M. Recker

RFC 1179
Line printer daemon protocol L. McLaughlin

RFC 1180
TCP/IP tutorial T. Socolofsky, C. Kale

RFC 1183
New DNS RR Definitions C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris

RFC 1184
Telnet Linemode Option D. Borman

RFC 1186
MD4 Message Digest Algorithm R.L. Rivest

RFC 1187
Bulk Table Retrieval with the SNMP M. Rose, K. McCloghrie, J. Davin

RFC 1188
Proposed Standard for the Transmission of IP Datagrams over FDDI Networks D. Katz

RFC 1190
Experimental Internet Stream Protocol: Version 2 (ST-II) C. Topolcic

RFC 1191
Path MTU discovery J. Mogul, S. Deering

RFC 1198
FYI on the X window system R. Scheifler

RFC 1207
FYI on Questions and Answers: Answers to commonly asked "experienced Internet user" questions G. Malkin, A. Marine, J. Reynolds

RFC 1208
Glossary of networking terms O. Jacobsen, D. Lynch

RFC 1213
Management Information Base for Network Management of TCP/IP-based internets: MIB-II K. McCloghrie, M.T. Rose

RFC 1215
Convention for defining traps for use with the SNMP M. Rose

RFC 1227
SNMP MUX protocol and MIB M.T. Rose

RFC 1228
SNMP-DPI: Simple Network Management Protocol Distributed Program Interface G. Carpenter, B. Wijnen

RFC 1229
Extensions to the generic-interface MIB K. McCloghrie

RFC 1230
IEEE 802.4 Token Bus MIB K. McCloghrie, R. Fox

RFC 1231
IEEE 802.5 Token Ring MIB K. McCloghrie, R. Fox, E. Decker

RFC 1236
IP to X.121 address mapping for DDN L. Morales, P. Hasse

RFC 1256
ICMP Router Discovery Messages S. Deering, Ed.

RFC 1267
Border Gateway Protocol 3 (BGP-3) K. Lougheed, Y. Rekhter

RFC 1268
Application of the Border Gateway Protocol in the Internet Y. Rekhter, P. Gross

RFC 1269
Definitions of Managed Objects for the Border Gateway Protocol: Version 3 S. Willis, J. Burruss

RFC 1270
SNMP Communications Services F. Kastenholz, ed.

RFC 1285
FDDI Management Information Base J. Case

RFC 1315
Management Information Base for Frame Relay DTEs C. Brown, F. Baker, C. Carvalho

RFC 1321
The MD5 Message-Digest Algorithm R. Rivest

RFC 1323
TCP Extensions for High Performance V. Jacobson, R. Braden, D. Borman

RFC 1325
FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions G. Malkin, A. Marine

RFC 1327
Mapping between X.400 (1988)/ISO 10021 and RFC 822 S. Hardcastle-Kille

RFC 1340
Assigned Numbers J. Reynolds, J. Postel

RFC 1344
Implications of MIME for Internet Mail Gateways N. Bornstein

RFC 1349
Type of Service in the Internet Protocol Suite P. Almquist

RFC 1350
The TFTP Protocol (Revision 2) K.R. Sollins

RFC 1351
SNMP Administrative Model J. Davin, J. Galvin, K. McCloghrie

RFC 1352
SNMP Security Protocols J. Galvin, K. McCloghrie, J. Davin

RFC 1353
Definitions of Managed Objects for Administration of SNMP Parties K. McCloghrie, J. Davin, J. Galvin

RFC 1354
IP Forwarding Table MIB F. Baker

RFC 1356
Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode A. Malis, D. Robinson, R. Ullmann

RFC 1358
Charter of the Internet Architecture Board (IAB) L. Chapin

RFC 1363
A Proposed Flow Specification C. Partridge

RFC 1368
Definition of Managed Objects for IEEE 802.3 Repeater Devices D. McMaster, K. McCloghrie

RFC 1372
Telnet Remote Flow Control Option C. L. Hedrick, D. Borman

RFC 1374
IP and ARP on HIPPI J. Renwick, A. Nicholson

RFC 1381
SNMP MIB Extension for X.25 LAPB D. Throop, F. Baker

RFC 1382
SNMP MIB Extension for the X.25 Packet Layer D. Throop

RFC 1387
RIP Version 2 Protocol Analysis G. Malkin

RFC 1388
RIP Version 2 Carrying Additional Information G. Malkin

RFC 1389
RIP Version 2 MIB Extensions G. Malkin, F. Baker

RFC 1390
Transmission of IP and ARP over FDDI Networks D. Katz

RFC 1393
Traceroute Using an IP Option G. Malkin

RFC 1398
Definitions of Managed Objects for the Ethernet-Like Interface Types F. Kastenholz

RFC 1408
Telnet Environment Option D. Borman, Ed.

RFC 1413
Identification Protocol M. St. Johns

RFC 1416
Telnet Authentication Option D. Borman, ed.

RFC 1420
SNMP over IPX S. Bostock

RFC 1428
Transition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME G. Vaudreuil

RFC 1442
Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1443
Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1445
Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2) J. Galvin, K. McCloghrie

RFC 1447
Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2) K. McCloghrie, J. Galvin

RFC 1448
Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1464
Using the Domain Name System to Store Arbitrary String Attributes R. Rosenbaum

RFC 1469
IP Multicast over Token-Ring Local Area Networks T. Pusateri

RFC 1483
Multiprotocol Encapsulation over ATM Adaptation Layer 5 Juha Heinanen

RFC 1514
Host Resources MIB P. Grillo, S. Waldbusser

RFC 1516
Definitions of Managed Objects for IEEE 802.3 Repeater Devices D. McMaster, K. McCloghrie

RFC 1521
MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies N. Borenstein, N. Freed

RFC 1535
A Security Problem and Proposed Correction With Widely Deployed DNS Software E. Gavron

RFC 1536
Common DNS Implementation Errors and Suggested Fixes A. Kumar, J. Postel, C. Neuman, P. Danzig, S. Miller

RFC 1537
Common DNS Data File Configuration Errors P. Beertema

RFC 1540
Internet Official Protocol Standards J. Postel

RFC 1571
Telnet Environment Option Interoperability Issues D. Borman

RFC 1572
Telnet Environment Option S. Alexander

RFC 1573
Evolution of the Interfaces Group of MIB-II K. McCloghrie, F. Kastenholz

RFC 1577
Classical IP and ARP over ATM M. Laubach

RFC 1583
OSPF Version 2 J. Moy

RFC 1591
Domain Name System Structure and Delegation J. Postel

RFC 1592
Simple Network Management Protocol Distributed Protocol Interface Version 2.0 B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

RFC 1594
FYI on Questions and Answers— Answers to Commonly Asked "New Internet User" Questions A. Marine, J. Reynolds, G. Malkin

RFC 1644
T/TCP — TCP Extensions for Transactions Functional Specification R. Braden

RFC 1646
TN3270 Extensions for LUname and Printer Selection C. Graves, T. Butts, M. Angel

RFC 1647
TN3270 Enhancements B. Kelly

RFC 1652
SMTP Service Extension for 8bit-MIMEtransport J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker

RFC 1664
Using the Internet DNS to Distribute RFC1327 Mail Address Mapping Tables C. Allochio, A. Bonito, B. Cole, S. Giordano, R. Hagens

RFC 1693
An Extension to TCP: Partial Order Service T. Connolly, P. Amer, P. Conrad

RFC 1695
Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2 M. Ahmed, K. Tesink

RFC 1701
Generic Routing Encapsulation (GRE) S. Hanks, T. Li, D. Farinacci, P. Traina

RFC 1702
Generic Routing Encapsulation over IPv4 networks S. Hanks, T. Li, D. Farinacci, P. Traina

RFC 1706
DNS NSAP Resource Records B. Manning, R. Colella

RFC 1712
DNS Encoding of Geographical Location C. Farrell, M. Schulze, S. Pleitner D. Baldoni

RFC 1713
Tools for DNS debugging A. Romao

RFC 1723
RIP Version 2—Carrying Additional Information G. Malkin

RFC 1752
The Recommendation for the IP Next Generation Protocol S. Bradner, A. Mankin

RFC 1766
Tags for the Identification of Languages H. Alvestrand

RFC 1771
A Border Gateway Protocol 4 (BGP-4) Y. Rekhter, T. Li

RFC 1794
DNS Support for Load Balancing T. Brisco

RFC 1819
Internet Stream Protocol Version 2 (ST2) Protocol Specification—Version ST2+ L. Delgrossi, L. Berger Eds.

RFC 1826
IP Authentication Header R. Atkinson

RFC 1828
IP Authentication using Keyed MD5 P. Metzger, W. Simpson

RFC 1829
The ESP DES-CBC Transform P. Karn, P. Metzger, W. Simpson

RFC 1830
SMTP Service Extensions for Transmission of Large and Binary MIME Messages G. Vaudreuil

RFC 1831
RPC: Remote Procedure Call Protocol Specification Version 2 R. Srinivasan

RFC 1832
XDR: External Data Representation Standard R. Srinivasan

RFC 1833
Binding Protocols for ONC RPC Version 2 R. Srinivasan

RFC 1850
OSPF Version 2 Management Information Base F. Baker, R. Coltun

RFC 1854
SMTP Service Extension for Command Pipelining N. Freed

RFC 1869
SMTP Service Extensions J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker

RFC 1870
SMTP Service Extension for Message Size Declaration J. Klensin, N. Freed, K. Moore

RFC 1876
A Means for Expressing Location Information in the Domain Name System C. Davis, P. Vixie, T. Goodwin, I. Dickinson

RFC 1883
Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden

RFC 1884
IP Version 6 Addressing Architecture R. Hinden, S. Deering, Eds.

RFC 1886
DNS Extensions to support IP version 6 S. Thomson, C. Huitema

RFC 1888
OSI NSAPs and IPv6 J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd

RFC 1891
SMTP Service Extension for Delivery Status Notifications K. Moore

RFC 1892
The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages G. Vaudreuil

RFC 1894
*An Extensible Message Format for Delivery Status Notifications*K. Moore, G. Vaudreuil

RFC 1901
Introduction to Community-based SNMPv2 J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1902
Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1903
Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1904
Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1905
Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1906
Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1907
Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1908
Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1912
Common DNS Operational and Configuration Errors D. Barr

RFC 1918
Address Allocation for Private Internets Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear

RFC 1928
SOCKS Protocol Version 5 M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones

RFC 1930
Guidelines for creation, selection, and registration of an Autonomous System (AS) J. Hawkinson, T. Bates

RFC 1939
Post Office Protocol-Version 3 J. Myers, M. Rose

RFC 1981
Path MTU Discovery for IP version 6 J. McCann, S. Deering, J. Mogul

RFC 1982
Serial Number Arithmetic R. Elz, R. Bush

RFC 1985
SMTP Service Extension for Remote Message Queue Starting J. De Winter

RFC 1995
Incremental Zone Transfer in DNS M. Ohta

RFC 1996
A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY) P. Vixie

RFC 2010
Operational Criteria for Root Name Servers B. Manning, P. Vixie

RFC 2011
SNMPv2 Management Information Base for the Internet Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2012
SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2013
SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2018
TCP Selective Acknowledgement Options M. Mathis, J. Mahdavi, S. Floyd, A. Romanow

RFC 2026
The Internet Standards Process — Revision 3 S. Bradner

RFC 2030
Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI D. Mills

RFC 2033
Local Mail Transfer Protocol J. Myers

RFC 2034
*SMTP Service Extension for Returning Enhanced Error Codes*N. Freed

RFC 2040
*The RC5, RC5–CBC, RC-5–CBC-Pad, and RC5–CTS Algorithms*R. Baldwin, R. Rivest

RFC 2045
Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies N. Freed, N. Borenstein

RFC 2052
A DNS RR for specifying the location of services (DNS SRV) A. Gulbrandsen, P. Vixie

RFC 2065
Domain Name System Security Extensions D. Eastlake 3rd, C. Kaufman

RFC 2066
TELNET CHARSET Option R. Gellens

RFC 2080
RIPng for IPv6 G. Malkin, R. Minnear

RFC 2096
IP Forwarding Table MIB F. Baker

RFC 2104
HMAC: Keyed-Hashing for Message Authentication H. Krawczyk, M. Bellare, R. Canetti

RFC 2119
Keywords for use in RFCs to Indicate Requirement Levels S. Bradner

RFC 2133
Basic Socket Interface Extensions for IPv6 R. Gilligan, S. Thomson, J. Bound, W. Stevens

RFC 2136
Dynamic Updates in the Domain Name System (DNS UPDATE) P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound

RFC 2137
Secure Domain Name System Dynamic Update D. Eastlake 3rd

RFC 2163
Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM) C. Allocchio

RFC 2168
Resolution of Uniform Resource Identifiers using the Domain Name System R. Daniel, M. Mealling

RFC 2178
OSPF Version 2 J. Moy

RFC 2181
Clarifications to the DNS Specification R. Elz, R. Bush

RFC 2205
Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin

RFC 2210
The Use of RSVP with IETF Integrated Services J. Wroclawski

RFC 2211
Specification of the Controlled-Load Network Element Service J. Wroclawski

RFC 2212
Specification of Guaranteed Quality of Service S. Shenker, C. Partridge, R. Guerin

RFC 2215
General Characterization Parameters for Integrated Service Network Elements S. Shenker, J. Wroclawski

RFC 2217
Telnet Com Port Control Option G. Clarke

RFC 2219
Use of DNS Aliases for Network Services M. Hamilton, R. Wright

RFC 2228
FTP Security Extensions M. Horowitz, S. Lunt

RFC 2230
Key Exchange Delegation Record for the DNS R. Atkinson

RFC 2233
The Interfaces Group MIB using SMIv2 K. McCloghrie, F. Kastenholz

RFC 2240
A Legal Basis for Domain Name Allocation O. Vaughn

RFC 2246
The TLS Protocol Version 1.0 T. Dierks, C. Allen

RFC 2251
Lightweight Directory Access Protocol (v3) M. Wahl, T. Howes, S. Kille

RFC 2253
Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names M. Wahl, S. Kille, T. Howes

RFC 2254
The String Representation of LDAP Search Filters T. Howes

RFC 2261
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 2262
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 2271
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 2273
SNMPv3 Applications D. Levi, P. Meyer, B. Stewartz

RFC 2274
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 2275
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 2279
UTF-8, a transformation format of ISO 10646 F. Yergeau

RFC 2292
Advanced Sockets API for IPv6 W. Stevens, M. Thomas

RFC 2308
Negative Caching of DNS Queries (DNS NCACHE) M. Andrews

RFC 2317
Classless IN-ADDR.ARPA delegation H. Eidnes, G. de Groot, P. Vixie

RFC 2320
Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB) M. Greene, J. Luciani, K. White, T. Kuo

RFC 2328
OSPF Version 2 J. Moy

RFC 2345
Domain Names and Company Name Retrieval J. Klensin, T. Wolf, G. Oglesby

RFC 2352
A Convention for Using Legal Names as Domain Names O. Vaughn

RFC 2355
TN3270 Enhancements B. Kelly

RFC 2358
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson

RFC 2373
IP Version 6 Addressing Architecture R. Hinden, S. Deering

RFC 2374
An IPv6 Aggregatable Global Unicast Address Format R. Hinden, M. O'Dell, S. Deering

RFC 2375
IPv6 Multicast Address Assignments R. Hinden, S. Deering

RFC 2385
Protection of BGP Sessions via the TCP MD5 Signature Option A. Hefferman

RFC 2389
Feature negotiation mechanism for the File Transfer Protocol P. Hethmon, R. Elz

RFC 2401
Security Architecture for Internet Protocol S. Kent, R. Atkinson

RFC 2402
IP Authentication Header S. Kent, R. Atkinson

RFC 2403
The Use of HMAC-MD5–96 within ESP and AH C. Madson, R. Glenn

RFC 2404
The Use of HMAC-SHA–1–96 within ESP and AH C. Madson, R. Glenn

RFC 2405
The ESP DES-CBC Cipher Algorithm With Explicit IV C. Madson, N. Doraswamy

RFC 2406
IP Encapsulating Security Payload (ESP) S. Kent, R. Atkinson

RFC 2407
*The Internet IP Security Domain of Interpretation for ISAKMP*D. Piper

RFC 2408
Internet Security Association and Key Management Protocol (ISAKMP) D. Maughan, M. Schertler, M. Schneider, J. Turner

RFC 2409
The Internet Key Exchange (IKE) D. Harkins, D. Carrel

RFC 2410
The NULL Encryption Algorithm and Its Use With IPsec R. Glenn, S. Kent,

RFC 2428
FTP Extensions for IPv6 and NATs M. Allman, S. Ostermann, C. Metz

RFC 2445
Internet Calendaring and Scheduling Core Object Specification (iCalendar) F. Dawson, D. Stenerson

RFC 2459
Internet X.509 Public Key Infrastructure Certificate and CRL Profile R. Housley, W. Ford, W. Polk, D. Solo

RFC 2460
Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden

RFC 2461
Neighbor Discovery for IP Version 6 (IPv6) T. Narten, E. Nordmark, W. Simpson

RFC 2462
IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten

RFC 2463
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

RFC 2464
Transmission of IPv6 Packets over Ethernet Networks M. Crawford

RFC 2466
Management Information Base for IP Version 6: ICMPv6 Group D. Haskin, S. Onishi

RFC 2476
Message Submission R. Gellens, J. Klensin

RFC 2487
SMTP Service Extension for Secure SMTP over TLS P. Hoffman

RFC 2505
Anti-Spam Recommendations for SMTP MTAs G. Lindberg

RFC 2523
Photuris: Extended Schemes and Attributes P. Karn, W. Simpson

RFC 2535
Domain Name System Security Extensions D. Eastlake 3rd

RFC 2538
Storing Certificates in the Domain Name System (DNS) D. Eastlake 3rd, O. Gudmundsson

RFC 2539
Storage of Diffie-Hellman Keys in the Domain Name System (DNS) D. Eastlake 3rd

RFC 2540
Detached Domain Name System (DNS) Information D. Eastlake 3rd

RFC 2554
SMTP Service Extension for Authentication J. Myers

RFC 2570
Introduction to Version 3 of the Internet-standard Network Management Framework J. Case, R. Mundy, D. Partain, B. Stewart

RFC 2571
An Architecture for Describing SNMP Management Frameworks B. Wijnen, D. Harrington, R. Presuhn

RFC 2572
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 2573
SNMP Applications D. Levi, P. Meyer, B. Stewart

RFC 2574
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 2575
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 2576
Co-Existence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework R. Frye, D. Levi, S. Routhier, B. Wijnen

RFC 2578
Structure of Management Information Version 2 (SMIv2) K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2579
Textual Conventions for SMIv2 K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2580
Conformance Statements for SMIv2 K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2581
TCP Congestion Control M. Allman, V. Paxson, W. Stevens

RFC 2583
Guidelines for Next Hop Client (NHC) Developers R. Carlson, L. Winkler

RFC 2591
Definitions of Managed Objects for Scheduling Management Operations D. Levi, J. Schoenwaelder

RFC 2625
IP and ARP over Fibre Channel M. Rajagopal, R. Bhagwat, W. Rickard

RFC 2635
Don't SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam)* S. Hambridge, A. Lunde

RFC 2637
Point-to-Point Tunneling Protocol K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, G. Zorn

RFC 2640
Internationalization of the File Transfer Protocol B. Curtin

RFC 2665
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson

RFC 2671
Extension Mechanisms for DNS (EDNS0) P. Vixie

RFC 2672
Non-Terminal DNS Name Redirection M. Crawford

RFC 2675
IPv6 Jumbograms D. Borman, S. Deering, R. Hinden

RFC 2710
Multicast Listener Discovery (MLD) for IPv6 S. Deering, W. Fenner, B. Haberman

RFC 2711
IPv6 Router Alert Option C. Partridge, A. Jackson

RFC 2740
OSPF for IPv6 R. Coltun, D. Ferguson, J. Moy

RFC 2753
A Framework for Policy-based Admission Control R. Yavatkar, D. Pendarakis, R. Guerin

RFC 2782
A DNS RR for specifying the location of services (DNS SRV) A. Gubrandsen, P. Vixix, L. Esibov

RFC 2821
Simple Mail Transfer Protocol J. Klensin, Ed.

RFC 2822
Internet Message Format P. Resnick, Ed.

RFC 2840
TELNET KERMIT OPTION J. Altman, F. da Cruz

RFC 2845
Secret Key Transaction Authentication for DNS (TSIG) P. Vixie, O. Gudmundsson, D. Eastlake 3rd, B. Wellington

RFC 2851
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 2852
Deliver By SMTP Service Extension D. Newman

RFC 2874
DNS Extensions to Support IPv6 Address Aggregation and Renumbering M. Crawford, C. Huitema

RFC 2915
The Naming Authority Pointer (NAPTR) DNS Resource Record M. Mealling, R. Daniel

RFC 2920
SMTP Service Extension for Command Pipelining N. Freed

RFC 2930
Secret Key Establishment for DNS (TKEY RR) D. Eastlake, 3rd

RFC 2941
Telnet Authentication Option T. Ts'o, ed., J. Altman

RFC 2942
Telnet Authentication: Kerberos Version 5 T. Ts'o

RFC 2946
Telnet Data Encryption Option T. Ts'o

RFC 2952
Telnet Encryption: DES 64 bit Cipher Feedback T. Ts'o

RFC 2953
Telnet Encryption: DES 64 bit Output Feedback T. Ts'o

RFC 2992
Analysis of an Equal-Cost Multi-Path Algorithm C. Hopps

RFC 3019
IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol B. Haberman, R. Worzella

RFC 3060
Policy Core Information Model—Version 1 Specification B. Moore, E. Ellesson, J. Strassner, A. Westerinen

RFC 3152
Delegation of IP6.ARPA R. Bush

RFC 3164
The BSD Syslog Protocol C. Lonvick

RFC 3207
SMTP Service Extension for Secure SMTP over Transport Layer Security P. Hoffman

RFC 3226
DNSSEC and IPv6 A6 aware server/resolver message size requirements O. Gudmundsson

RFC 3291
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 3363
Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System R. Bush, A. Durand, B. Fink, O. Gudmundsson, T. Hain

RFC 3376
: *Internet Group Management Protocol, Version 3* B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan

RFC 3390
: *Increasing TCP's Initial Window* M. Allman, S. Floyd, C. Partridge

RFC 3410
: *Introduction and Applicability Statements for Internet-Standard Management Framework* J. Case, R. Mundy, D. Partain, B. Stewart

RFC 3411
: *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks* D. Harrington, R. Presuhn, B. Wijnen

RFC 3412
: *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)* J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 3413
: *Simple Network Management Protocol (SNMP) Applications* D. Levi, P. Meyer, B. Stewart

RFC 3414
: *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* U. Blumenthal, B. Wijnen

RFC 3415
: *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* B. Wijnen, R. Presuhn, K. McCloghrie

RFC 3416
: *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)* R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3417
: *Transport Mappings for the Simple Network Management Protocol (SNMP)* R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3418
: *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)* R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3419
: *Textual Conventions for Transport Addresses* M. Daniele, J. Schoenwaelder

RFC 3484
: *Default Address Selection for Internet Protocol version 6 (IPv6)* R. Draves

RFC 3493
: *Basic Socket Interface Extensions for IPv6* R. Gilligan, S. Thomson, J. Bound, J. McCann, W. Stevens

RFC 3513
: *Internet Protocol Version 6 (IPv6) Addressing Architecture* R. Hinden, S. Deering

RFC 3526
: *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)* T. Kivinen, M. Kojo

RFC 3542
Advanced Sockets Application Programming Interface (API) for IPv6 W. Richard Stevens, M. Thomas, E. Nordmark, T. Jinmei

RFC 3566
The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec S. Frankel, H. Herbert

RFC 3569
An Overview of Source-Specific Multicast (SSM) S. Bhattacharyya, Ed.

RFC 3584
Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework R. Frye, D. Levi, S. Routhier, B. Wijnen

RFC 3602
The AES-CBC Cipher Algorithm and Its Use with IPsec S. Frankel, R. Glenn, S. Kelly

RFC 3629
UTF-8, a transformation format of ISO 10646 R. Kermode, C. Vicisano

RFC 3658
Delegation Signer (DS) Resource Record (RR) O. Gudmundsson

RFC 3678
Socket Interface Extensions for Multicast Source Filters D. Thaler, B. Fenner, B. Quinn

RFC 3715
IPsec-Network Address Translation (NAT) Compatibility Requirements B. Aboba, W. Dixon

RFC 3810
Multicast Listener Discovery Version 2 (MLDv2) for IPv6 R. Vida, Ed., L. Costa, Ed.

RFC 3826
The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model U. Blumenthal, F. Maino, K McCloghrie.

RFC 3947
Negotiation of NAT-Traversal in the IKE T. Kivinen, B. Swander, A. Huttunen, V. Volpe

RFC 3948
UDP Encapsulation of IPsec ESP Packets A. Huttunen, B. Swander, V. Volpe, L. DiBurro, M. Stenberg

RFC 4001
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 4007
IPv6 Scoped Address Architecture S. Deering, B. Haberman, T. Jinmei, E. Nordmark, B. Zill

RFC 4022
Management Information Base for the Transmission Control Protocol (TCP) R. Raghunarayan

RFC 4106
The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) J. Viega, D. McGrew

RFC 4109
Algorithms for Internet Key Exchange version 1 (IKEv1) P. Hoffman

RFC 4113
Management Information Base for the User Datagram Protocol (UDP) B. Fenner, J. Flick

RFC 4191
Default Router Preferences and More-Specific Routes R. Draves, D. Thaler

RFC 4217
Securing FTP with TLS P. Ford-Hutchinson

RFC 4292
IP Forwarding Table MIB B. Haberman

RFC 4293
Management Information Base for the Internet Protocol (IP) S. Routhier

RFC 4301
Security Architecture for the Internet Protocol S. Kent, K. Seo

RFC 4302
IP Authentication Header S. Kent

RFC 4303
IP Encapsulating Security Payload (ESP) S. Kent

RFC 4304
Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP) S. Kent

RFC 4307
Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2) J. Schiller

RFC 4308
Cryptographic Suites for IPsec P. Hoffman

RFC 4434
The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol P. Hoffman

RFC 4443
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

RFC 4552
Authentication/Confidentiality for OSPFv3 M. Gupta, N. Melam

RFC 4678
Server/Application State Protocol v1 A. Bivens

RFC 4753
ECP Groups for IKE and IKEv2 D. Fu, J. Solinas

RFC 4754
IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA) D. Fu, J. Solinas

RFC 4809
Requirements for an IPsec Certificate Management Profile C. Bonatti, Ed., S. Turner, Ed., G. Lebovitz, Ed.

RFC 4835
Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) V. Manral

RFC 4862
IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten, T. Jinmei

RFC 4868
Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec S. Kelly, S. Frankel

RFC 4869
Suite B Cryptographic Suites for IPsec L. Law, J. Solinas

RFC 4941
Privacy Extensions for Stateless Address Autoconfiguration in IPv6 T. Narten, R. Draves, S. Krishnan

RFC 4945
The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX B. Korver

RFC 5014
IPv6 Socket API for Source Address Selection E. Nordmark, S. Chakrabarti, J. Laganier

RFC 5095
Deprecation of Type 0 Routing Headers in IPv6 J. Abley, P. Savola, G. Neville-Neil

RFC 5175
IPv6 Router Advertisement Flags Option B. Haberman, Ed., R. Hinden

RFC 5282
Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol D. Black, D. McGrew

RFC 5996
Internet Key Exchange Protocol Version 2 (IKEv2) C. Kaufman, P. Hoffman, Y. Nir, P. Eronen

Internet drafts

Internet drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Other groups can also distribute working documents as Internet drafts. You can see Internet drafts at http://www.ietf.org/ID.html.

Appendix G. Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, you can view the information through the z/OS Internet Library website or IBM Knowledge Center. If you continue to experience problems, send an email to mhvrcfs@us.ibm.com or write to:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. See z/OS TSO/E Primer, z/OS TSO/E User's Guide, and z/OS ISPF User's Guide Vol I for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer or Library Server versions of z/OS books in the Internet library at www.ibm.com/systems/z/os/zos/bkserv/.

One exception is command syntax that is published in railroad track format, which is accessible using screen readers with IBM Knowledge Center, as described in "Dotted decimal syntax diagrams."

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing IBM Knowledge Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should see separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- A question mark (?) means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- An exclamation mark (!) means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted

decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

- An asterisk (*) means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  Notes:

  1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.
- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the USA.

IBM may not offer all of 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want 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.

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.

If you are viewing this information softcopy, the photographs and color illustrations might not appear.

COPYRIGHT LICENSE:

This information might contain 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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_.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California. Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Portions of this publication relating to RPC are Copyright © Sun Microsystems, Inc., 1988, 1989.

Some portions of this publication relating to X Window System** are Copyright © 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, and the Massachusetts Institute Of Technology, Cambridge, Massachusetts.

Some portions of this publication relating to X Window System are Copyright © 1986, 1987, 1988 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute the M.I.T., Digital Equipment Corporation, and Hewlett-Packard Corporation portions of this software and its documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T., Digital, and Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T., Digital, and Hewlett-Packard make no representation about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright © 1983, 1995-1997 Eric P. Allman

Copyright © 1988, 1993 The Regents of the University of California.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

> This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software program contains code, and/or derivatives or modifications of code originating from the software program "Popper." Popper is Copyright ©1989-1991 The Regents of the University of California. Popper was created by Austin Shelton, Information Systems and Technology, University of California, Berkeley.

Permission from the Regents of the University of California to use, copy, modify, and distribute the "Popper" software contained herein for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies. HOWEVER, ADDITIONAL PERMISSIONS MAY BE NECESSARY FROM OTHER PERSONS OR ENTITIES, TO USE DERIVATIVES OR MODIFICATIONS OF POPPER.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THE POPPER SOFTWARE, OR ITS DERIVATIVES OR MODIFICATIONS, AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE POPPER SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Copyright © 1983 The Regents of the University of California.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com)

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

X Window System is a trademark of The Open Group.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: http://www-01.ibm.com/software/support/systemsz/lifecycle/
- For information about currently-supported IBM hardware, contact your IBM representative.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java-based trademarks are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Adobe and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Bibliography

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

z/OS Communications Server documentation is available in the following forms:

- Online at the z/OS Internet Library web page at www.ibm.com/systems/z/os/zos/bkserv/
- In softcopy on CD-ROM collections. See "Softcopy information" on page xviii.

z/OS Communications Server library updates

An index to z/OS Communications Server book updates is at http://www.ibm.com/support/docview.wss?uid=swg21178966. Updates to documents are also available on RETAIN® and in information APARs (info APARs). Go to http://www.ibm.com/software/network/commserver/zos/support to view information APARs.

z/OS Communications Server information

z/OS Communications Server product information is grouped by task in the following tables.

Planning

Title	Number	Description
z/OS Communications Server: New Function Summary	GC27-3664	This document is intended to help you plan for new IP or SNA function, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC27-3663	This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server's support of IPv6, coexistence with IPv4, and migration issues.

Resource definition, configuration, and tuning

Title	Number	Description
z/OS Communications Server: IP Configuration Guide	SC27-3650	This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document with the z/OS Communications Server: IP Configuration Reference.

Title	Number	Description
z/OS Communications Server: IP Configuration Reference	SC27-3651	This document presents information for people who want to administer and maintain IP. Use this document with the z/OS Communications Server: IP Configuration Guide. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC27-3672	This document presents the major concepts involved in implementing an SNA network. Use this document with the z/OS Communications Server: SNA Resource Definition Reference.
z/OS Communications Server: SNA Resource Definition Reference	SC27-3675	This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document with the z/OS Communications Server: SNA Network Implementation Guide.
z/OS Communications Server: SNA Resource Definition Samples	SC27-3676	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: IP Network Print Facility	SC27-3658	This document is for systems programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Operation

Title	Number	Description
z/OS Communications Server: IP User's Guide and Commands	SC27-3662	This document describes how to use TCP/IP applications. It contains requests with which a user can log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC27-3661	This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
z/OS Communications Server: SNA Operation	SC27-3673	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SC27-3665	This document contains essential information about SNA and IP commands.

Customization

Title	Number	Description
z/OS Communications Server: SNA Customization	SC27-3666	This document enables you to customize SNA, and includes the following information: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines

Writing application programs

Title	Number	Description
z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference	SC27-3660	This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC27-3649	This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS® using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC27-3653	This document is for programmers who want application programs that use the IMS™ TCP/IP application development services provided by the TCP/IP Services of IBM.
z/OS Communications Server: IP Programmer's Guide and Reference	SC27-3659	This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC27-3674	This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC27-3669	This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC27-3670	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.
z/OS Communications Server: CSM Guide	SC27-3647	This document describes how applications use the communications storage manager.

Title	Number	Description
z/OS Communications Server: CMIP Services and Topology Agent Guide	SC27-3646	This document describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The document provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis

Title	Number	Description
z/OS Communications Server: IP Diagnosis Guide	GC27-3652	This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: ACF/TAP Trace Analysis Handbook	GC27-3645	This document explains how to gather the trace data that is collected and stored in the host processor. It also explains how to use the Advanced Communications Function/Trace Analysis Program (ACF/TAP) service aid to produce reports for analyzing the trace data information.
z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures and z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT	GC27-3667 GC27-3668	These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
z/OS Communications Server: SNA Data Areas Volume 1 and z/OS Communications Server: SNA Data Areas Volume 2	GC31-6852 GC31-6853	These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and codes

Title	Number	Description
z/OS Communications Server: SNA Messages	SC27-3671	This document describes the ELM, IKT, IST, IUT, IVT, and USS messages. Other information in this document includes: • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
z/OS Communications Server: IP Messages Volume 1 (EZA)	SC27-3654	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)	SC27-3655	This volume contains TCP/IP messages beginning with EZB or EZD.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC27-3656	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)	SC27-3657	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC27-3648	This document describes codes and other information that appear in z/OS Communications Server messages.

Index

Communicating your comments to IBM

If you especially like or dislike anything about this document, you can send us comments electronically by using one of the following methods:

Internet email:
comsvrcf@us.ibm.com

World Wide Web:
http://www.ibm.com/systems/z/os/zos/webqs.html

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

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

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

Product Number: 5650-ZOS

Printed in USA

SC27-3661-02